GAMING MACHINE

RELATED APPLICATIONS

This application claims the priority of Japanese Patent Application Nos. 11-365707 filed on December 22, 1999, 11-373997 filed on December 28, 1999 and 2000-021130 filed on January 31, 2000, which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a gaming machine, and more particularly, to a gaming machine which performs electrical control on the basis of a hierarchical processing structure.

Description of the Related Art

Slot machines have hitherto been known as gaming machines of this type.

A related-art slot machine comprises: three reels for variably displaying a plurality of types of symbols required for game; a start switch for commencing spinning of all the reels in unison on condition that a gaming token has been inserted; a stop switch for stopping spinning of the reels individually; a hopper for paying out gaming tokens as a prize if the reels constitute a predetermined combination when stopped; a liquid-crystal display device for displaying game information; and a speaker for generating sounds associated with the game.

In such a related-art slot machine, a player actuates a start switch after a game starting condition has been properly set, whereupon a plurality of reels on which symbols are printed rotate. When the player actuates stop switches assigned to respective reels, the reels stop rotating. The plurality of symbols printed on the surfaces of the reels are displayed stationarily.

When the stationarily-displayed symbols enter a predetermined winning mode, gaming tokens are paid out from the hopper as a prize. Provided that the stationarily-displayed symbols enter a predetermined jackpot winning mode, the

player is allowed to play a special game (e.g., a big bonus game or regular bonus game) which is more advantageous than a regular game. Such a round of game operations is controlled by a control unit, such as a microcomputer.

In order to cause a slot machine to provide a variety of presentations, such display of animated images, control operation to be performed by the control unit has recently become more complicated, thereby posing difficulty for a single control unit to control all game operations. For this reason, the control unit is divided into a main control unit and sub-control units, and there has recently been employed a method of controlling a slot machine by means of a hierarchical control architecture. For instance, a determination pertaining to player's profits is rendered by a main control unit, and the sub-control units control a hopper, a liquid-crystal display device, and a speaker.

In a case where a liquid-crystal device provides visual presentations, great processing capability is required for effecting image processing, and an enormous amount of image data must be stored. For these reasons, if a control unit is divided into a main control unit and sub-control units such that a determination pertaining to player's profits is rendered by the main control unit and such that image processing which is lower-hierarchical-level processing of the main control unit is performed by the sub-control units, as is the case with the related-art slot machine, burdens imposed on the sub-control units increase, thereby incurring the risk of precluding smooth control of game.

In order to attract a player to play a game in accordance with his tastes, there must be prepared a plurality of types of slot machines which provide different presentations. Even slot machines providing different presentations have a commonality with regard to the principal control of game operation.

However, in the related-art slot machine, a control processing architecture is not fragmented. For this reason, a control unit must be re-designed for each of slot machines performing different game operations, thus adding to manufacturing costs.

Even when a portion of functions of the control unit have failed, the entire control unit must be replaced. In this regard, costs of a slot machine are also increased.

The present invention has been conceived in light of the foregoing circumstances and is aimed at providing a gaming machine capable of diminishing manufacturing costs and maintenance costs, as well as capable of performing smooth control operation even when providing a variety of gaming presentations.

SUMMARY OF THE INVENTION

To these ends, a gaming machine according to the present invention comprises:

a variable display section for variably displaying a plurality of types of symbols required for gaming;

a starting device for starting variable display of the symbols;

a stopping device for stopping the symbols being variably displayed;

a profit provision device for providing a player with profits if a combination of symbols constitutes a predetermined winning mode when the symbols are stopped and displayed;

an image display device for displaying information about a game;

a sound generation device for generating a sound relevant to a game; and

a control device for electrically controlling the variable display section, the starting device, the stopping device, the profit provision device, the image display device, and the sound generation device on the basis of a hierarchical processing structure, the control device including

a main control device for controlling the first hierarchical level, which is the highest hierarchical processing level of the gaming machine,

an intermediate control device for controlling the second hierarchical level situated lower than the first hierarchical level, under control of the main control device, and

a lower control device for controlling a third hierarchical level situated lower than the second hierarchical level, under control of the intermediate control device.

Preferably, the control device performs processing for making a decision pertaining to the player's profits, and the lower control device controls image display to be performed by the image display device. Further, the intermediate control device preferably performs control operations other than those to be performed by the lower control device.

Preferably, the intermediate control device performs control operations including a sound generation control operation to be performed by the sound generation device.

Preferably, the intermediate control means is controlled in accordance with a command code transmitted from the main control means, and the lower control means is controlled in accordance with a command code transmitted from the intermediate control means.

Preferably, the lower control device comprises a plurality of control boards specialized for specific control processing operations, respectively.

Preferably, the lower control means is equipped with a lamp control board.

Preferably, the lower control means is equipped with a control board having mounted thereon at least an image control CPU and character ROM.

The control device is embodied by, for example, a microcomputer having features such as a CPU, ROM, and RAM. The gaming machine is controlled by means of the CPU operating in accordance with a sequence program stored in the ROM or a like memory device.

Further, the present invention provides a gaming machine comprising:

variable display means for variably displaying a plurality of types of symbols required for gaming;

starting means for starting variable display of the symbols;

stopping means for stopping the symbols being variably displayed;

profit provision means for providing a player with profits if a combination of symbols constitutes a predetermined winning mode when the symbols are stopped and displayed;

image display means for displaying information about a game;

sound generation means for generating a sound relevant to a game; and

control means for electrically controlling the variable display means, the starting means, the stopping means, the profit provision means, the image display means, and the sound generation means on the basis of a hierarchical processing structure, the control means including

main control means for controlling the first hierarchical level, which is the highest hierarchical processing level of the gaming machine,

intermediate control means for controlling the second hierarchical level situated lower than the first hierarchical level, under control of the main control means, and

lower control means for controlling a third hierarchical level situated lower than the second hierarchical level, under control of the intermediate control means.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 a front view of a slot machine according to an embodiment of the present invention;

Fig. 2 is a schematic block diagram of a control unit (main control board);

Fig. 3 is a schematic block diagram of the control unit (a sub-control board and an image control board);

Fig. 4 is a table describing an IN port (input port [0]) constituting the sub-control board;

Fig. 5 is a table describing an OUT port (output port [0]) constituting the sub-control board;

Fig. 6 is a table describing an IN port (a command receive port [of higher hierarchical level]) constituting the sub-control board;

Fig. 7 is a table describing an IN port (a command receive port [of lower hierarchical level]) constituting the sub-control board;

Fig. 8 is a table describing an OUT port (a command receive port [of lower hierarchical level]) constituting the sub-control board;

Fig. 9 is a table describing an OUT port (a sound output port) constituting the sub-control board;

Fig. 10 is a table for describing commands to be received by the command receive port;

Fig. 11 is an illustration for describing a data value corresponding to a command code (MCMD_INIT);

Fig. 12 is an illustration for describing a data value corresponding to a command code (MCMD_DEMO);

Fig. 13 is an illustration for describing a data value corresponding to a command code (MCMD_MDIN);

Fig. 14 is an illustration for describing a data value corresponding to a command code (MCMD_NMST);

Fig. 15 is an illustration for describing a data value corresponding to a command code (MCMD_RBST);

Fig. 16 is an illustration for describing a data value corresponding to a command code (MCMD_BBST);

Fig. 17 is an illustration for describing a data value corresponding to a command code (MCMD_RLSP);

Fig. 18 is an illustration for describing a data value corresponding to a command code (MCMD_NHIT);

Fig. 19 is an illustration for describing a data value corresponding to a command code (MCMD_JHIT);

Fig. 20 is an illustration for describing a data value corresponding to a command code (MCMD_POFN);

Fig. 21 is an illustration for describing a data value corresponding to a command code (MCMD_BNST);

Fig. 22 is an illustration for describing a data value corresponding to a command code (MCMD_BBFN);

Fig. 23 is an illustration for describing a data value corresponding to a command code (MCMD_ERR);

Fig. 24 is an illustration for describing a data value corresponding to a command code (MCMD_PSEL);

Fig. 25 is an illustration for describing a data value corresponding to a command code (MCMD_SUND);

Fig. 26 is a table for describing a display control code to be transmitted from a liquid crystal display command transmission port;

Fig. 27 is an illustration for describing a data value corresponding to a command code (DSP_INIT);

Fig. 28 is an illustration for describing a data value corresponding to a command code (DSP_DEMO);

Fig. 29 is an illustration for describing a data value corresponding to a command code (DSP_REEL);

Fig. 30 is an illustration for describing displayed symbol data;

Fig. 31 is an illustration for describing a data value corresponding to a command code (DSP_NSTR);

Fig. 32 is an illustration for describing a data value corresponding to a command code (DSP_SSTR);

Fig. 33 is an illustration for describing a data value corresponding to a command code (DSP_NLSP);

Fig. 34 is an illustration for describing a data value corresponding to a

command code (DSP_RECH);

Fig. 35 is an illustration for describing a data value corresponding to a command code (DSP_SSTP);

Fig. 36 is an illustration for describing a data value corresponding to a command code (DSP_NHIT);

Fig. 37 is an illustration for describing a data value corresponding to a command code (DSP_SHIT);

Fig. 38 is an illustration for describing a data value corresponding to a command code (DSP_BHIT);

Fig. 39 is an illustration for describing a data value corresponding to a command code (DSP_BSTG);

Fig. 40 is an illustration for describing a data value corresponding to a command code (DSP_RSTR);

Fig. 41 is an illustration for describing a data value corresponding to a command code (DSP_JHIT);

Fig. 42 is an illustration for describing a data value corresponding to a command code (DSP_BSTR);

Fig. 43 is an illustration for describing a data value corresponding to a command code (DSP_BNHT);

Fig. 44 is an illustration for describing a data value corresponding to a command code (DSP_BRHT);

Fig. 45 is an illustration for describing a data value corresponding to a command code (DSP_ERR);

Fig. 46 is a table for describing constants to be used in control processing performed by the sub-control board;

Fig. 47 is an illustration for describing a timer to be used in the control processing performed by the sub-control board;

Fig. 48 is an illustration for describing a flag to be used in the control

processing performed by the sub-control board;

Fig. 49 is a table for describing a work area used in the control processing to be performed by the sub-control board;

Fig. 50 is a table for describing a work area used in the control processing to be performed by the sub-control board;

Fig. 51 is an illustration for describing a sequence control table stored in program ROM of the sub-control board;

Fig. 52 is an illustration for describing command data to be transmitted to sound source IC;

Fig. 53 is an illustration for describing sound effects (powerball 3 *LI-ZHI* failure) generated by the sound source IC;

Fig. 54 is an illustration for describing sound effects (powerball 3, *LI-ZHI* winning) generated by the sound source IC;

Fig. 55 is an illustration for describing sound effects (RB stage 3, completion of eight jackpot winnings) generated by the sound source IC;

Fig. 56 is an illustration for describing sound effects (RB stage 3, jackpot failures ended with punctures) generated by the sound source IC;

Fig. 57 is an illustration for describing sound effects (RB stage 3, jackpot winnings ended with punctures) generated by the sound source IC;

Fig. 58 is a table for describing sound output request control code;

Fig. 59 is a table for describing sound output request control code;

Fig. 60 is a table for describing sound output request control code;

Fig. 61 is an illustration for describing a sound output data table;

Fig. 62 is an illustration for describing a sound output data table;

Fig. 63 is an illustration for describing a sound output data table;

Fig. 64 is an illustration for describing a sound output data table;

Fig. 65 is an illustration for describing a sound output data table;

Fig. 66 is an illustration for describing a sound output data table;

Fig. 67 is an illustration for describing a sound output data table;

Fig. 68 is an illustration for describing a sound output data table;

Fig. 69 is an illustration for describing a sound output data table;

Fig. 70 is an illustration for describing a sound output data table;

Fig. 71 is an illustration for describing a sound output data table;

Fig. 72 is an illustration for describing a sound output data table;

Fig. 73 is an illustration for describing a sound output data table;

Fig. 74 is an illustration for describing a sound output data table;

Fig. 75 is an illustration for describing a sound output data table;

Fig. 76 is a table for describing sound codes used in the sound output data table;

Fig. 77 is an illustration for describing a *LI-ZHI* presentation selection table to be used during a normal game;

Fig. 78 is an illustration for describing the *LI-ZHI* presentation selection table to be used during a normal game;

Fig. 79 is an illustration for describing the *LI-ZHI* presentation selection table to be used during a normal game;

Fig. 80 is an illustration for describing the *LI-ZHI* presentation selection table to be used during a normal game;

Fig. 81 is an illustration for describing the *LI-ZHI* presentation selection table to be used during a normal game;

Fig. 82 is an illustration for describing a *LI-ZHI* presentation selection table to be used during an internally-generated bonus game;

Fig. 83 is an illustration for describing the *LI-ZHI* presentation selection table to be used during an internally-generated bonus game;

Fig. 84 is an illustration for describing the *LI-ZHI* presentation selection table to be used during an internally-generated bonus game;

Fig. 85 is an illustration for describing the *LI-ZHI* presentation selection

table to be used during an internally-generated bonus game;

Fig. 86 is an illustration for describing the *LI-ZHI* presentation selection table to be used during an internally-generated bonus game;

Fig. 87 is an illustration for describing a *LI-ZHI* presentation selection table to be used for illuminating WIN lamps;

Fig. 88 is an illustration for describing the *LI-ZHI* presentation selection table to be used for illuminating WIN lamps;

Fig. 89 is an illustration for describing the *LI-ZHI* presentation selection table to be used for illuminating WIN lamps;

Fig. 90 is an illustration for describing the *LI-ZHI* presentation selection table to be used for illuminating WIN lamps;

Fig. 91 is an illustration for describing the *LI-ZHI* presentation selection table to be used for illuminating WIN lamps;

Fig. 92 is an illustration for describing a *LI-ZHI* presentation displayed-symbol table for effecting "BBRECHDATBB" generation presentation;

Fig. 93 is an illustration for describing a *LI-ZHI* presentation displayed-symbol table for effecting "RBRECHDATRB" generation presentation;

Fig. 94 is an illustration for describing a *LI-ZHI* presentation displayed-symbol table for effecting "MSRECCHDAT" failure presentation;

Fig. 95 is an illustration for describing a table for selecting a center symbol to be displayed when "balancing-on-rolling-ball *LI-ZHI*" has failed;

Fig. 96 is an illustration for describing a table for selecting symbols to be displayed during a normal game (no *LI-ZHI*);

Fig. 97 is an illustration for describing a table for selecting symbols to be displayed during an internally-generated bonus game (without *LI-ZHI*);

Fig. 98 is an illustration for describing a table for selecting flash data to be used with the *LI-ZHI* presentation selection table;

Fig. 99 is a table for describing the relationship between flash data, random

numbers for selection, a reel blinking pattern, and presentation sound (start sound);

Fig. 100 is a table for describing the relationship between flash data, random numbers for selection, a reel blinking pattern, and presentation sound (start sound);

Fig. 101 is a table for describing the relationship between flash data, random numbers for selection, a reel blinking pattern, and presentation sound (start sound);

Fig. 102 is a table for describing the relationship between flash data, random numbers for selection, a reel blinking pattern, and presentation sound (start sound);

Fig. 103 is a demonstration display appearing in response to a command code "02h" (shown in Fig. 28);

Fig. 104 is a demonstration display appearing in response to the command code "02h" (shown in Fig. 28);

Fig. 105 is a demonstration display appearing in response to the command code "02h" (shown in Fig. 28);

Fig. 106 is a demonstration display appearing in response to the command code "02h" (shown in Fig. 28);

Fig. 107 is a reel screen display appearing in response to a command code "03h" (shown in Fig. 29);

Fig. 108 is a reel screen display appearing in response to the command code "03h" (shown in Fig. 29);

Fig. 109 is a sign presentation display appearing in response to a command code "04h" (shown in Fig. 31);

Fig. 110 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 111 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 112 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 113 is a sign presentation display appearing in response to the command

code "04h" (shown in Fig. 31);

Fig. 114 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 115 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 116 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 117 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 118 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 119 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 120 is a sign presentation display appearing in response to the command code "04h" (shown in Fig. 31);

Fig. 121 is a *LI-ZHI* presentation display appearing in response to a command code "07h" (shown in Fig. 34);

Fig. 122 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 123 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 124 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 125 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 126 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 127 is a *LI-ZHI* presentation display appearing in response to the

command code "07h" (shown in Fig. 34);

Fig. 128 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 129 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 130 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 131 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 132 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 133 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 134 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 135 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 136 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 137 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 138 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 139 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 140 is a *LI-ZHI* presentation display appearing in response to the command code "07h" (shown in Fig. 34);

Fig. 141 is a *LI-ZHI* presentation display appearing in response to the

command code "07h" (shown in Fig. 34);

Fig. 142 is a big bonus stage presentation screen appearing in response to a command code "0Ch" (shown in Fig. 39);

Fig. 143 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 144 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 145 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 146 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 147 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 148 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 149 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 150 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 151 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 152 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 153 is a big bonus stage presentation screen appearing in response to the command code "0Ch" (shown in Fig. 39);

Fig. 154 is a regular bonus stage presentation screen appearing in response to a command code "0Dh" (shown in Fig. 40);

Fig. 155 is a regular bonus stage presentation screen appearing in response

to the command code "0Dh" (shown in Fig. 40);

Fig. 156 is a regular bonus stage presentation screen appearing in response to the command code "0Dh" (shown in Fig. 40);

Fig. 157 is a regular bonus stage presentation screen appearing in response to the command code "0Dh" (shown in Fig. 40);

Fig. 158 is a jackpot-game-winning presentation screen appearing in response to a command code "0Eh" (shown in Fig. 41);

Fig. 159 is a jackpot-game-winning presentation screen appearing in response to the command code "0Eh" (shown in Fig. 41);

Fig. 160 is a jackpot-game-winning presentation screen appearing in response to the command code "0Eh" (shown in Fig. 41);

Fig. 161 is a big bonus game presentation screen appearing in response to a command code "0Fh" (shown in Fig. 42);

Fig. 162 is a big bonus game presentation screen appearing in response to the command code "0Fh" (shown in Fig. 42);

Fig. 163 is a big bonus game presentation screen appearing in response to the command code "0Fh" (shown in Fig. 42);

Fig. 164 is a small-jackpot combination winning presentation screen appearing during a normal game in big bonus in response to a command code "10h" (shown in Fig. 43);

Fig. 165 is a small-jackpot combination winning presentation screen appearing during a normal game in big bonus in response to the command code "10h" (shown in Fig. 43);

Fig. 166 is a small-jackpot combination winning presentation screen appearing during a normal game in big bonus in response to the command code "10h" (shown in Fig. 43);

Fig. 167 is a small-jackpot combination winning presentation screen appearing during a normal game in big bonus in response to the command code

"10h" (shown in Fig. 43);

Fig. 168 is a small-jackpot combination winning presentation screen appearing during a normal game in big bonus in response to the command code "10h" (shown in Fig. 43);

Fig. 169 is a small-jackpot combination winning presentation screen appearing during a normal game in big bonus in response to the command code "10h" (shown in Fig. 43);

Fig. 170 is a regular bonus winning presentation screen appearing during a big bonus in response to a command code "11h" (shown in Fig. 44);

Fig. 171 is a regular bonus winning presentation screen appearing during a big bonus in response to the command code "11h" (shown in Fig. 44);

Fig. 172 is a regular bonus winning presentation screen appearing during a big bonus in response to the command code "11h" (shown in Fig. 44);

Fig. 173 is a regular bonus winning presentation screen appearing during a big bonus in response to the command code "11h" (shown in Fig. 44);

Fig. 174 is an error display screen appearing in response to a command code "12h" (shown in Fig. 45);

Fig. 175 is an error display screen appearing in response to the command code "12h" (shown in Fig. 45);

Fig. 176 is an error display screen appearing in response to the command code "12h" (shown in Fig. 45);

Fig. 177 is an error display screen appearing in response to the command code "12h" (shown in Fig. 45);

Fig. 178 is an error display screen appearing in response to the command code "12h" (shown in Fig. 45);

Fig. 179 is a flowchart showing procedures for receive interrupt processing;

Fig. 180 is a flowchart showing procedures for receive interrupt processing;

Fig. 181 is a flowchart showing procedures for receive interrupt processing;

Fig. 182 is a flowchart showing procedures for receive interrupt processing;

Fig. 183 is a flowchart showing procedures for timer interrupt processing;

Fig. 184 is a flowchart showing procedures for receive interrupt processing;

Fig. 185 is a flowchart showing procedures for receive interrupt processing;

Fig. 186 is a flowchart showing procedures for transmission start processing;

Fig. 187 is a flowchart showing procedures for command transmission processing;

Fig. 188 is a flowchart showing procedures for presentation restoration check processing;

Fig. 189 is a flowchart showing procedures for presentation restoration check processing;

Fig. 190 is a flowchart showing procedures for presentation nonrestoration processing;

Fig. 191 is a flowchart showing procedures for presentation control processing;

Fig. 192 is a flowchart showing procedures for presentation control processing;

Fig. 193 is a flowchart showing procedures for game status check processing;

Fig. 194 is a flowchart showing procedures for game status check processing;

Fig. 195 is a flowchart showing procedures for game status check processing;

Fig. 196 is a flowchart showing procedures for presentation sequence control processing;

Fig. 197 is a flowchart showing procedures for presentation sequence control processing;

Fig. 198 is a flowchart showing procedures for analyzing a received command;

Fig. 199 is a flowchart showing procedures for demonstration display command processing;

Fig. 200 is a flowchart showing procedures for token insertion command processing;

Fig. 201 is a flowchart showing procedures for token insertion command processing;

Fig. 202 is a flowchart showing procedures for game start command processing during a normal game;

Fig. 203 is a flowchart showing procedures for reel spinning start command processing during a regular bonus game;

Fig. 204 is a flowchart showing procedures for reel spinning start command processing during a regular bonus game;

Fig. 205 is a flowchart showing procedures for reel spinning start command processing during a big bonus game;

Fig. 206 is a flowchart showing procedures for reel spinning start command processing during a big bonus game;

Fig. 207 is a flowchart showing procedures for stop reel command processing;

Fig. 208 is a flowchart showing procedures for stop reel command processing;

Fig. 209 is a flowchart showing procedures for stop reel command processing;

Fig. 210 is a flowchart showing procedures for stop reel command processing;

Fig. 211 is a flowchart showing procedures for winning (all-reel-stop) command processing;

Fig. 212 is a flowchart showing procedures for winning (all-reel-stop) command processing;

Fig. 213 is a flowchart showing procedures for winning (all-reel-stop) command processing;

Fig. 214 is a flowchart showing procedures for jackpot-winning command processing;

Fig. 215 is a flowchart showing procedures for jackpot-winning command processing;

Fig. 216 is a flowchart showing procedures for payout completion command processing;

Fig. 217 is a flowchart showing procedures for processing pertaining to an instruction changing the status of a bonus game;

Fig. 218 is a flowchart showing procedures for processing pertaining to an instruction changing the status of a bonus game;

Fig. 219 is a flowchart showing procedures for operation command processing to be performed when a big bonus ends;

Fig. 220 is a flowchart showing procedures for operation command processing to be performed when a big bonus ends;

Fig. 221 is a flowchart showing procedures for error presentation command processing;

Fig. 222 is a flowchart showing procedures for error presentation command processing;

Fig. 223 is a flowchart showing procedures for main-CPU-presentation-type command processing;

Fig. 224 is a flowchart showing procedures for main-CPU-presentation-type command processing;

Fig. 225 is a flowchart showing procedures for main-CPU-presentation-type command processing;

Fig. 226 is a flowchart showing procedures for sound single command processing;

Fig. 227 is a flowchart showing procedures for presentation selection processing to be performed when a normal game and a bonus game is internally

generated;

Fig. 228 is a flowchart showing procedures for presentation selection processing to be performed when a normal game and a bonus game is internally generated;

Fig. 229 is a flowchart showing procedures for presentation selection processing to be performed when a normal game and a bonus game is internally generated;

Fig. 230 is a flowchart showing procedures for presentation selection processing to be performed when a normal game and a bonus game is internally generated;

Fig. 231 is a flowchart showing procedures for bonus-winning processing;

Fig. 232 is a flowchart showing procedures for bonus-winning processing;

Fig. 233 is a flowchart showing procedures for winning during a big bonus game;

Fig. 234 is a flowchart showing procedures for winning during a big bonus game;

Fig. 235 is a flowchart showing procedures for winning during a big bonus game;

Fig. 236 is a flowchart showing procedures for winning during a big bonus game;

Fig. 237 is a flowchart showing procedures for presentation status reset processing;

Fig. 238 is a flowchart showing procedures for sound restoration processing;

Fig. 239 is a flowchart showing procedures for received command storage processing;

Fig. 240 is a flowchart showing procedures for command fetch processing;

Fig. 241 is a flowchart showing procedures for winning-sign-presentation-type selection processing;

Fig. 242 is a flowchart showing procedures for winning-sign-presentation-type selection processing;

Fig. 243 is a flowchart showing procedures for *LI-ZHI*-presentation-type selection processing;

Fig. 244 is a flowchart showing procedures for *LI-ZHI*-presentation-type selection processing;

Fig. 245 is a flowchart showing procedures for *LI-ZHI*-presentation-type selection processing;

Fig. 246 is a flowchart showing procedures for selecting from a selection table symbols to be displayed;

Fig. 247 is a flowchart showing procedures for sound control processing;

Fig. 248 is a flowchart showing procedures for sound initialization processing;

Fig. 249 is a flowchart showing procedures for sound mute processing;

Fig. 250 is a flowchart showing procedures for sound mute processing;

Fig. 251 is a flowchart showing procedures for sound output processing;

Fig. 252 is a flowchart showing procedures for sound output processing;

Fig. 253 is a flowchart showing procedures for sound output processing;

Fig. 254 is a flowchart showing procedures for sound output processing;

Fig. 255 is a flowchart showing procedures for SD_OUT sound output data transmission processing;

Fig. 256 is a flowchart showing procedures for stopping replay of all channels;

Fig. 257 is a flowchart showing transmission timing of a transmission command;

Fig. 258 is a flowchart showing transmission timing of a transmission command;

Fig. 259 is a flowchart showing transmission timing of a transmission

command;

Fig. 260 is an illustration showing the configuration of a transmission command;

Fig. 261 is an illustration for describing a reel blinking pattern;

Fig. 262 is an illustration for describing a reel blinking pattern;

Fig. 263 is an illustration for describing a reel blinking pattern;

Fig. 264 is an illustration for describing a reel blinking pattern;

Fig. 265 is an illustration for describing a reel blinking pattern;

Fig. 266 is an illustration for describing a reel blinking pattern;

Fig. 267 is an illustration for describing a reel blinking pattern; and

Fig. 268 is an illustration for describing a reel blinking pattern.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will be described by the accompanying drawings.

Throughout the following descriptions, a slot machine is taken as a typical gaming machine.

Slot Machine

Fig. 1 is a front view showing an embodiment of the slot machine according to the present invention.

As shown in Fig. 1, the slot machine 1 according to the present invention has a housing 3, and a reclosable front door 2 is provided on the front surface of the housing 3. A display window 4 is formed at a position above and near the center of the front face of the front door 2. The respective outer peripheral surfaces of three reels 5a to 5c disposed within the housing 3 face the display window 4. Winning-line indicators 6 for indicating effective winning lines are printed across the display window 4 so as to extend to the peripheries thereof. Here, the winning-line indicators 6 are constituted of a total of five indicators; that is, three horizontal indicators and two diagonal indicators crossing the horizontal ones. Also,

effective-line indicator lamps 7a to 7e are provided on the left-end side of the winning-line indicators 6a to 6e for indicating corresponding effective winning lines.

In the following descriptions, the left-side reel 5a is called a first reel; the right-side reel 5c is called a second reel; and the center reel 5b is called a third reel. Stoppage of the first reel is called a first stoppage; stoppage of the second reel is called a second stoppage; and stoppage of the third reel is called a third stoppage. A symbol displayed at the time of stoppage of the first reel is sometimes called simply a left-side displayed symbol; a symbol displayed at the time of stoppage of the second reel is sometimes called simply a right-side displayed symbol; and a symbol displayed at the time of stoppage of the third reel is sometimes called simply a center displayed symbol.

Although the three reels 5a to 5c are disposed within a single display window 4 in the embodiment shown in Fig. 1, discrete display windows may be provided so as to correspond to the reels 5a to 5c. Further, although five effective winning lines are provided here, the effective winning lines may be provided in any number, such as seven or nine.

On the right-end side of the winning-line indicators 6, there are provided a replay display lamp 8 which is illuminated at the time of a replay winning; a game stop display lamp 9 for displaying a wait time from starting of the last reel spinning until starting of the next reel spinning is permitted; a winning display lamp 10 for indicating that a winning combination is generated; a game start display lamp 11 for indicating that a start switch 20 is active; and a gaming token insertion lamp 12 for indicating that insertion of gaming tokens is permitted.

An image display section 13 for displaying game information is disposed in substantially the center of the front face of the front door 2 and below the display window 4. The image display section 13 consists of, for example, a liquid-crystal display, a plasma display, an EL display, a CRT display, a dot-matrix indicator, or the like, and can display a game presentation, which will be explained later in detail, in

the form of a stationary image or an animation.

In the embodiment shown in FIG. 1, the reels 5a to 5c, which are variable display means, and the image display section 13 are constructed separately from each other. However, they may be indicated in a single display device through use of a CRT display of about 17 inches or a like device. In this case, simulated reels can be displayed on the CRT in place of the reels 5a to 5c.

A token insertion slot 14 is formed on the right side of the image display section 13 for enabling simultaneous insertion of gaming tokens to be used for gaming. An open/close key 15 is provided on the front door 2 and in a position lower right with reference to the token insertion slot 14. A first bet switch 16 is provided on the left side of the image display section 3 for enabling insertion of gaming tokens on a one-by-one basis within a credited range. A second bet switch 17 is provided next the first bet switch 16 for insertion of two gaming tokens within the credited range. In addition, a max bet switch 18 is provided just to the left of the image display section 13 for enabling insertion of gaming tokens up to a maximum bet number (three in this embodiment) within the credited range.

Though not shown, a card unit may be attached to the slot machine 1 so as to form a card-type slot machine which accepts a valued medium such as a prepaid card or the like for loaning out gaming tokens, so that credit can be made in a credit section as in the case where the gaming tokens are inserted into the slot machine 1. Further, the card-type slot machine may be configured such that, while the card unit is attached to the slot machine 1, a hopper of the slot machine main body is actuated upon a gaming token loan-out operation, so as to loan out a predetermined number of gaming tokens to a token tray 22.

Below the image display section 13, the front face of the front door 2 is provided with a C/P switch 19 for changeover between credit and payout of the tokens acquired by the player; the start switch 20 for starting spinning of each of the reels 5a to 5c on condition that a gaming token has been inserted; and three stop

switches 21a to 21c for stopping spinning of the respective reels 5a to 5c.

The token tray 22 is provided in a lower part of the front door 2 for receiving gaming tokens paid out as a prize. A token payout slot 23 is provided opposite the token tray 22. Sound transmission holes 24 are provided on the front face of the front door 2 and in a position above the token tray 22. A speaker 25 is provided within the housing 3 so as to face the sound transmission holes 24.

An upper part of the front face of the housing 3 is provided with a dividend display section 26 for displaying the number of dividend gaming tokens to be paid out as a reward for a winning combination. A decorating section 27 provided with a plurality of decorating lamps (not shown) is provided above the dividend display section 26. The decorating section 27 is divided into a plurality of (e.g., 8) lateral sub-sections, each sub-section including a decorating lamp. The decorating lamp is illuminated or blinked according to the status of a game, thus enhancing the entertainment value of a game.

Within the housing 3, the reels 5a to 5c are rotatably disposed at their respective positions where their outer peripheral surfaces face the display window 4. Further, a hopper (not shown) is disposed at a position communicating with the token payout slot 23 for paying out gaming tokens as a prize. A control unit (not shown) is disposed within the housing 3 for electrically controlling the slot machine 1.

A light-transparent reel tape having a plurality of kinds of symbols printed thereon at predetermined intervals is affixed onto the outer peripheral surface of each of the reels 5a to 5c. For example, the kinds of symbols include "7," "EXTRA," "CHERRY," "DIAMOND," "DRAGON," "BALL," etc. Twenty-one symbols are printed on each of the reels 5a to 5c. When symbols constitute a combination "7-7-7," a big bonus (BB) is started. When symbols constitute a combination "EXTRA-EXTRA-EXTRA," a regular bonus (RB) is started. In this situation, when symbol "CHERRY" is stopped at the middle stage of the left reel 5a, two gaming

tokens are paid out. When symbol "CHERRY" is stopped at either the upper or lower stage of the left reel 5a, four gaming tokens are paid out. When symbols constitute a combination "DIAMOND-DIAMOND-DIAMOND", ten gaming tokens are paid out, and when symbols constitute a combination "DRAGON-DRAGON-DRAGON", seven gaming tokens are paid out. When symbols constitute a combination "DRAGON-EXTRA-EXTRA" (i.e., an internally-generated dragon combination to be described later), three gaming tokens are paid out. When symbols constitute a combination "BALL-BALL-BALL," the player can replay a game.

Here, the kinds of symbols and the number of symbols displayed in each of the reels 5a to 5c can be changed as appropriate. For example, the kinds of symbols may include, in addition to those mentioned above, "BELL," "ORANGE," "PERSON," "ANIMAL," "FISH," "JAC," "BAR," and the like. Further, each of symbols may be painted in a plurality of colors such that the symbols are distinguishable from each other.

Disposed inside each of the reels 5a to 5c are three back lamps (not shown) in a vertical row for illuminating from inside the respective reel 5a to 5c in a transmitting manner the symbols seen through the display window 4. As the back lamps are illuminated, each of the reels 5a to 5c is illuminated from inside, whereby symbols stationarily displayed on effective winning lines can be highlighted.

<Game Played on Slot Machine>

In order to play a game with the slot machine 1, the player inserts gaming tokens into the token insertion slot 14. Alternatively, the player actuates any of the bet switches 16, 17, and 18, to thereby insert gaming tokens for game within a credited range. Now, effective winning lines are determined according to the number of inserted gaming tokens, and corresponding ones of the effective-line indicator lamps 7a to 7e are illuminated. For example, when one gaming token is inserted, a middle horizontal line becomes effective. When two gaming tokens are

inserted, three horizontal lines; that is, the upper, middle, and lower horizontal lines, become effective. When three gaming tokens, which constitute the maximum permissible bet number, are inserted, five lines in total consisting of three horizontal lines in the upper, middle, and lower parts and two diagonal lines become effective.

Subsequently, when the player actuates the start switch 20, all the reels 5a to 5c start spinning simultaneously, whereby a plurality of kinds of symbols printed on the respective outer peripheral surfaces of the reels 5a to 5c are displayed while vertically moving within the display window 4. When spinning of the reels 5a to 5c reaches a predetermined speed, corresponding stop switches 21a to 21c are made active. As the player actuates the stop switches 21a to 21c, the corresponding reels 5a to 5c stop spinning.

In a case where the combination of the symbols stationarily displayed on an effective winning line enters a predetermined winning mode, the number of gaming tokens corresponding to this winning mode are paid out as a prize or added as a credit.

<Winning Mode>

Predetermined winning modes include normal winning modes, and special winning modes to become a starting condition for special games which are more advantageous to the player than normal games. Further, the special winning modes include so-called big bonus and so-called regular bonus.

Specific examples of winning modes include, as frequently-occurring winning modes in general, so-called small-JACKPOT combinations such as "CHERRY," "DIAMOND," "DRAGON," and "WATERMELON," and winning modes known as replay by which the next game can be played without insertion of a gaming token.

Usually, if no winning occurs in an internally-generated game, the established internally-generated combination will not be transferred to the next and later games. Further, a relatively small number of gaming tokens; i.e., 15 or fewer tokens, are paid out. Big bonuses and regular bonuses, which are called special

winning modes are generated when combinations of predetermined special game starting symbols are aligned on the winning lines. Here, the big bonuses include CT-attached big bonuses, wherein the CT-attached big bonuses further include games known as challenge time (CT) in which stop control for the reels 5a to 5c by random number sampling is stopped for a predetermined period after completion of a big bonus game.

In such special winning games called big bonuses and regular bonuses, winning modes occur at a higher probability than in normal games. The internal sampling probability of these winning modes is much lower than that of small-jackpot combinations or replay winning. Hence, even when no winning occurs in the internally-generated game; that is, when no special winning mode constituted on a winning line, their internally-generated jackpots are arranged so as to be transferable to the next and later games. A large number of gaming tokens; i.e., on the order of about 350 to 450 for a big bonus and about 120 for a regular bonus, can be acquired in one jackpot game.

Internal generation is that the control unit performs random number sampling and enables occurrence of a winning mode on the basis of a result of sampling. If internal generation is performed, the reels 5a to 5c are regulated to stop so as to constitute winning modes to the greatest extent practicable when the player performs an operation for stopping the stop switches 21a to 21c. In contrast, if internal generation is not performed, even when the player performs an operation for stopping the stop switches 21a to 21c in an attempt to constitute a winning mode, control is effected so as not to constitute a winning mode.

<Big Bonuses>

The winning modes of big bonus are concerned with games which are started on condition that the combination of symbols stationarily displayed on an effective winning line assumes "7," "7," and "7." In this case, after a predetermined number of; e.g., 15 gaming tokens, have been paid out, the player can play a big bonus game

in which the player can advantageously acquire a greater number of gaming tokens than in a normal game.

In this big bonus game, games similar to normal games with an increased sampling probability of small-jackpot combinations (called normal games in a big bonus (BB) game [normal games in BB]) can be played up to 30 times. During the normal games in BB, if the symbols stationarily displayed on an effective winning line assume a combination of, for example, "DIAMOND," "DIAMOND," and "DIAMOND," ten gaming tokens are paid out. If "CHERRY" is stationarily displayed in a left-side position on the display window 4, two gaming tokens are paid out. If the symbols stationarily displayed on an effective winning line assume a combination of "BALL," "BALL," and "BALL," which is a specific winning mode, five gaming tokens are paid out, and the player is allowed to play a JAC game up to three times.

In terms of a combination of stopped symbols constituting a winning mode as well as the number of gaming tokens paid out when a winning mode is constituted, normal games in BB are performed under substantially the same conditions as those under which normal games are played outside a period of a big bonus, except that no replay winning mode is available and that there is a specific winning mode for shifting into a JAC game, in place of a specific winning mode serving as a starting condition of a special game.

<JAC Game>

In a JACKPOT game, the start switch 20 is actuated with insertion of a predetermined number of gaming tokens; e.g., one gaming token, to thereby start spinning the reels 5a to 5c. Subsequently, the stop switches 21a to 21c are actuated to stop spinning of the reels 5a to 5c.

The player can play a bonus game (legally called "Jackpot Game") in which, if a combination of stopped symbols constitutes a predetermined combination of, e.g., "BALL," "BALL," and "BALL," a predetermined number of; e.g., 15 gaming tokens, are paid out. In connection with the slot machine constructed so as to generate a

regular bonus, the number of gaming tokens to be paid during a regular bonus are substantially the same as those paid out during a JAC game arising during the big bonus. In some cases, a big bonus is defined as being constituted of a predetermined number of regular games in BB and a predetermined number of regular bonuses.

In the JAC game, the maximum number of games and the maximum number of wins are restricted. For example, when the above-mentioned game is played 12 times, which constitutes the maximum game number, or the number of the above-mentioned wins reaches eight, which is the maximum winning number, the JAC game ends.

<Regular Bonus>

In addition to the above-mentioned big bonus, there are special game modes known as so-called regular bonus.

The winning modes of regular bonus are concerned with games to be started on condition that symbols stationarily displayed on an effective winning line constitute a combination of "EXTRA," "EXTRA," and "EXTRA." After a predetermined number of; for example, fifteen gaming tokens have been paid out, the player can play regular bonus games.

In a regular bonus game, the acquirable profit is smaller than that in the above-mentioned big bonus game. For example, the above-mentioned JAC game is allowed to be played one time at maximum.

Control Unit

Gaming operations in the slot machine 1 are controlled by a control unit.

This control unit will be explained with reference to Figs. 2 and 3.

As shown in Figs. 2 and 3, the control unit comprises a main control board 100, a sub-control board 200, and an image control board 300.

As shown in Fig. 2, the main control board 100 performs main control of gaming operations in the slot machine. The main control board 100 comprises a

main CPU 101; ROM 102; RAM 103; a clock circuit 104 for generating an operating clock signal for the main CPU 101; and a probability setting section 105 for setting the probability of generation of big bonus and like bonuses.

The ROM 102 stores data items such as a winning probability table for determining the probability of sampling, internally-generated combinations, a stop control table for controlling stoppage of the reels 5a to 5c according to the status of gaming, or the like, as well as storing processing procedures for gaming in the slot machine 1 as a sequence program. As the main CPU 101 and other circuits operate according to the sequence program, gaming in the slot machine 60 is controlled.

The clock circuit 104 comprises a clock pulse generator 106 for generating a reference clock at a predetermined frequency, and a divider 107 for generating an operating clock signal for the CPU 101 by dividing the reference clock signal.

The probability setting section 105 comprises a random number generator 108 for generating random numbers within a predetermined range under the control of the main CPU 101, and a random number sampling circuit 109 for extracting a given random number from the random numbers generated in the random number generator 108 and transmitting the thus-extracted random number to the main CPU 101. A probability setting switch 110 for setting the probability of occurrence of big bonus is connected to the probability setting section 105.

The probability setting section 105 generates random numbers to be used for normal games, big bonus, or the like. Specifically, a determination as to whether or not there is an internally-generated combination and determination of a winning combination are made by means of comparing the random number extracted at the time of actuation of the start switch 20 with the winning probability table stored in the ROM 102. Internally-generated bonus combinations, which are the results of internal generation of big bonus or regular bonus, can be transferred to the next and later games. In contrast, internally-generated small-jackpot combination or replay winning is only valid in the game in which they are internally generated.

A plurality of I/O ports provided with the CPU 101 are connected to: the first bet switch 16; the second bet switch 17; the max bet switch 18; the C/P switch 19; the start switch 20; a token sensor 111 for detecting gaming tokens inserted from the token insertion slot 14; a payout setting switch 112 for determining whether to effect a payout process; a play-out cancel switch 113 for canceling the play-out status; a motor drive circuit 114; a reel-position detection circuit 115; a reel-stop-signal circuit 116; a hopper drive circuit 117; a payout-completion-signal circuit 118; a speaker drive circuit 119; and a lamp drive circuit 120.

In the following, the individual circuits mentioned above will be explained in detail.

The motor drive circuit 114 is connected to stepping motors 121a to 121c for spinning the respective reels 5a to 5c. Spinning of the reels 5a to 5c is stopped or started by means of enabling or disabling supply of drive pulses to the stepping motors 121a to 121c under control of the main CPU 101.

The reel-position detection circuit 115 is provided with a position detection sensor (not shown) comprising an optical sensor or the like for detecting the spinning position of each of the reels 5a to 5c. Position detection signals concerning the reels 5a to 5c detected by the position detection sensor are transmitted to the main CPU 101.

When a combination of symbols stationarily displayed on the respective reels 5a to 5c constitute a winning mode, profits are awarded to the player. The principal functions of the reels 5a to 5c are to "variably display visually-cognizable symbols or images, to determine variations in accordance with the player's operation, and to determine whether to reward profits to the player on the basis of the result of determination." Simulated reels, for example, as well as mechanical reels as employed in the present embodiment, may be variably displayed on an electrical image display device. In addition to variable display of a plurality of symbols, a single or plurality of characters or background images are variably displayed. A

determination as to whether to award profits to the player may be made on the basis of fixed statuses; for instance, the posture, expressions, and reactions, of the stopped characters or images; the relationship between characters or images; the extent to which background is changed; combinations of words; progress in a story; or the like.

The reel-stop-signal circuit 116 is connected to the stop switches 21a to 21c. As the player actuates the stop switches 21a to 21c, the actuation is detected, and the resultant stop switch detection signal is transmitted to the main CPU 101. More specifically, after the individual reels 5a to 5c attain constant-speed spinning as a result of actuation of the start switch 20, actuation of the stop switches 21a to 21c is allowed. When the stop switches 21a to 21c are pressed one after another, a total of seven frames extending to the fourth frame from the shortest stoppable position are checked according to the stop switch detection signal, the position detection signal, and the stop control table stored in the ROM 102. If there is any symbol corresponding to the internally-generated combination, there is performed so-called draw-in control operation such that the symbol is aligned on the effective winning line. Further, there is performed so-called kick control operation so as not to establish any other winning combinations which are not internally generated. Thus, the reels 5a to 5c are stopped. In the case of losing (also called a "failure" in drawings) in which there is no internally-generated combination, the individual reels 5a to 5c are stopped such that no internally-generated combination is established. In an internally-generated bonus game to which the internally-generated bonus combination has been transferred, internally-generated combinations other than the internally-generated bonus combination are sampled. If a small-jackpot combination or replay winning is internally generated, a draw-in control operation is performed so as to preferentially attain the small-jackpot combination or replay. The only requirement for means corresponding to the stop switches 21a to 21c is to be able to determine one from images that are being displayed variably and to

display the thus-determined image. The means may be embodied as a touch sensor, a device which produces a stop signal upon voice recognition, a thermally-sensitive sensor, or the like. There is not necessity for the means being fixed on the body of the slot machine. The means may be controlled remotely in a wireless or wired manner.

The hopper drive circuit 117 is connected to a hopper 122 for storing gaming tokens.

The payout-completion-signal circuit 118 is connected to a token storage section 123 and a token detecting section 124. The token storage section 123 stores the gaming tokens inserted from the token insertion slot 14 or gaming tokens to be paid out as a prize. The token storage section 123 can store gaming tokens until they reach a predetermined maximum permissible storage number of; for example, 50. Gaming tokens are stored in the hopper 122 in a number up to 50, and the 51st and later gaming tokens are actually paid out from the hopper 122 to the token tray 22. The actually paid-out tokens are counted by the token detecting section 124 at the time when being paid out from the hopper 122 to the token tray 22. In the operation of paying out gaming tokens at the time of winning, if the sum value to be added to and stored in the token storage section 123 or the counted value in the token detection section 124 reaches a predetermined payout number, a payout completion signal is transmitted from the payout-completion-signal circuit 118 to the main CPU 101.

Means for awarding profits to the player is not limited to a hopper and may be embodied by a prize payout device. For instance, in a case where Pachinko balls are employed as gaming media, a prize ball payout device can be provided. Further, the means for awarding profits to the player may not be a physical gaming medium. For instance, in the case of a card-type slot machine, a prize value may be written directly onto a card. If electronic money is utilized, profits may be transferred directly into the player's account.

The lamp drive circuit 120 is connected to back lamps 125 for illuminating from inside the reels 5a to 5c in a transmitting manner the symbols seen through the display window 4. Three back lamps 125 are provided in a vertical row for each of the reels 5a to 5c. Nine back lamps 125 in total illuminate the reels 5a to 5c from inside in a transmitting manner.

It is not necessarily required that the slot machine 1 have a housing. The slot machine 1 may be constructed so as to connect a main server having stored therein game programs with a terminal device assigned to each player by way of a communications link such as the Internet, and to enable the player to play a game by means of displaying the reels 5a to 5c along with visual presentations on a display provided on each terminal. In this case, a game program may be downloaded to each of terminal devices, and a game may be initiated on each terminal device. Alternatively, a program stored in the main server may be initiated directly from each terminal device. Moreover, terminal devices may be arranged so as to be interconnected by way of a communications link so that a game played on a certain terminal device can be remotely controlled from another terminal device.

<Sub-Control Board>

The sub-control board 200 transmits a signal pertaining to image presentation control to the image control board 300 under control of the main control board 100 and controls generation of sound effects.

As shown in Fig. 3, the sub-control board 200 is mounted with a sub-CPU 201, program ROM 202, and control RAM 203. The sub-CPU 201 on the sub-control board 200 receives a signal from the main control board 100 by way of an IN port 204. A command of this signal is transmitted in only one direction from the main control board 100 to the sub-control board 200, thereby preventing unauthorized manipulation of the main control board 100, which would otherwise be caused by loading an illegal program into the sub-control board 200. Further, the sub-CPU 201 determines various presentation images on the basis of the data output

from the main control board 100 and the selection table stored in the program ROM 202, and transmits a signal to the image control board 300 in only one direction by way of an OUT port 205. The sub-control board 200 is additionally mounted with a sound-source IC 206 which generates sound effects or the like from the speaker 25 by way of a power amplifier 207. In the present embodiment, the sub-CPU 201 has a receiving function of controlling receipt of a command from the main control board 100; a presentation selection function of selecting and determining presentations; a transmission function of transmitting the kind of thus selected presentation to the image control board 300; and a sound-effect control function of sequentially controlling specific sound effects according to the kind of selected presentation. The present embodiment primarily relates to the sub-control board 200. If a statement concerning sequence control is found in the following description, the term in principle expresses sequence control of sound effects. Since sequence control of images involves a heavy load, sequence control of specific images is delegated to the image control board 300 provided in a hierarchical level immediately below the sub-control board 200.

<Image Control Board>

The image control board 300 controls display of specific images on the image display section 13 under control of the sub-control board 200.

As shown in Fig. 3, the image control board 300 is mounted with an image control CPU 301; program ROM 302; control RAM 303; an image control IC 304; character ROM 305; and video RAM 306. The image control board 300 receives a signal from the sub-control board 200 by way of an IN port 307, and transmits a drive signal to the image control IC 304. Under the control of the image control CPU 301, the image control IC 304 receives signals from the character ROM 305 and video RAM 306, and controls the image display section 13 (e.g., a color liquid-crystal panel), thereby effecting image displays.

In the present embodiment, according to the general kind of presentations

selected by the sub-CPU 201, the image control CPU 301 performs sequence control of specific images.

The functions assigned to the control boards in the present embodiment may be changed. For instance, the sub-CPU 201 may be assigned only a receiving function and an presentation selection function, and sequence control of sound effects may be delegated to a separately sound-only CPU.

In a case where the sub-CPU 201 has high presentation or where the volume of images to be handled is small, an image sequence control function may be assigned to the sub-CPU 201. In addition, ramp control performed by the main control board 100 may be delegated to the sub-control board 200 (in this case, the sub-control board 200 is preferably mounted with a lamp control board, as is the image control board 300).

<Ports of Sub-Control Board>

A specific example of the IN port 204 and that of the OUT port 205 constituting the sub-control board 200 will now be described by reference to Figs. 4 through 9.

As shown in Figs. 4 through 9, each of ports has eight data terminals.

The IN port 204 is constituted of an input port [0] for receiving a strobe signal and power-down signal output from the main control board 100 (INMAP0 in Fig. 4); a command receipt port [of higher hierarchical level] for receiving data from the main control board 100 (COMHMAP in Fig. 6); and a command receipt port [of lower hierarchical level] (COMPLMAP in Fig. 7).

The OUT port 205 is constituted of a control output port [0] for outputting a sound-mute-control output signal, a liquid-crystal-strobe output signal, and a watchdog-timer reset signal (CNTMAP0 in Fig. 5); a liquid-crystal display command transmission port for outputting data to the image display CPU (LCDMAP in Fig. 8); and a sound output port for outputting data to the sound-source IC 206 (SUNDMAP in Fig. 9).

<Commands Transmitted from the Main Control Board to the Sub-Control Board>

As shown in Fig. 260, a command transmitted from the main control board 100 is constituted of four bytes in total; namely, preceding two-byte data having stored therein information about various games, and subsequent two-byte data having stored therein the status of gaming and a BCC (Block Check Character) value. The status of gaming is expressed in ascending sequence from lower bits: namely, whether or not RB operation is in progress (bit 0); whether or not BB winning is expected [i.e., the symbols displayed on the reels 5a, 5b, and 5c are now in *TEN P'AIS* status (in which two identical symbols are displayed on an effective line and in which, if the third symbol becomes identical with the other two symbols, a jackpot becomes effective, and this status will be hereinafter referred to simply as "*TEN P'AIS*" throughout the specification) (bit 1); whether or not replay is in progress (bit 2); whether or not the main control board 100 is in losing (bit 3); whether or not the slot machine is in a play-out status (bit 5); whether or not generation of bonus is determined and displayed (bit 6); and whether or not BB operation is in progress (bit 7). The subsequent two-byte data are common to all transmission commands, and hence the preceding two-byte data will be described.

Fig. 10 shows a command which is transmitted from the main control board 100 and is received by the command receipt port (shown in Figs. 6 and 7). More specifically, Fig. 10 explains command codes received by the command receipt port. The left-side column shows command codes; the center column shows data values; and the right-side column shows descriptions of the command codes.

Data pertaining to each of the command codes consist of one byte (= eight bits). Data pertaining to each command code will be described in more detail by reference to Figs. 11 through 25.

Figs. 11 through 25 show data values corresponding to typical command codes. Contents of bytes are described in descending sequence from the top of the

code.

Fig. 11 shows a data value corresponding to a command code (MCMD_INIT) shown in Fig. 10. Command code "01H" is described as a presentation initialization command. This command code is transmitted when the slot machine is powered up, settings are changed, or play is not restored for reasons of corruption of RAM. In terms of presentation, all sounds are muted, and a display appearing on a liquid-crystal screen disappears. By means of this presentation initiation command, a liquid-crystal display erasure command (01h) is selected and transmitted as a command (hereinafter called a sub-sub-command) to be transmitted to the image control board 300. A capital H is suffixed to a command code transmitted from the main control board 100 to the sub-control board 200, and a lower-case "h" is suffixed to a command code transmitted from the sub-control board 200 to the image control board 300.

Fig. 12 shows a data value corresponding to a command code (MCMD_DEMO) shown in Fig. 10, and a command code "02H" is described as a demonstration display command. When the player fails to perform an operation during a given period of time (e.g., 30 seconds), the main control board 100 transmits the demonstration display code (but does not transmit the code during a bonus game). A demonstration display, a dividend display, or explanations about game rules are displayed as presentations. By means of the demonstration display code, a demonstration display sub-sub-command (02h) is selected and transmitted.

Fig. 13 shows a data value corresponding to a command code (MCMD_MDIN) shown in Fig. 10, and a command code "03H" in the first byte is described as gaming token insertion data. Further, data pertaining to the number of gaming tokens to be inserted are stored in the second byte. The command code is transmitted when a gaming token is inserted. At the time of addition of credits, a sound single presentation instruction to be described later is issued. By means of the command code, a reel screen display sub-sub-command (03h) is selected and

transmitted.

Fig. 14 shows a data value corresponding to a command code (MCMD_NMST) shown in Fig. 10, and a command code "O4H" provided in the first byte is described as a data command for starting a game during a normal game. Data pertaining to the type of winning, such as a big bonus, a regular bonus, or a replay winning, are added to the second byte. During a normal game (including a normal game arising during a big bonus game), the command code is transmitted after the type of internally-generated winning has been determined through probability sampling processing upon detection of actuation of the start switch 20. The command code is followed by a main-CPU-presentation-type command or a command for starting spinning of reels during a big bonus, which will be described later. Changes in presentations on the slot machine are not induced by merely the game start data command code; presentations are determined on the basis of commands following the game start data command code (only the type of winning is transmitted).

Fig. 15 shows a data value corresponding to a command code (MCMD_RBST) shown in Fig. 10, and a command code "O5H" provided in the first byte is described as data for starting spinning of reels during a regular bonus game. Data pertaining to the number of available regular bonus games and data pertaining to the number of available regular bonus operations are added to the second byte. The command code is transmitted when spinning of the reels 5a to 5c is started upon detection of actuation of the start switch 20 during a regular bonus game. A game start sound and sound effects for presentation of a regular bonus game are generated as presentations. Further, regular-bonus-game presentations (of three types for each regular bonus stage) are displayed. By means of the command code, there is selected and transmitted a sub-sub-command (0Dh) for starting spinning of reels during a regular bonus game.

Fig. 16 shows a data value corresponding to a command code (MCMD_BBST)

shown in Fig. 10, and a command code "O6H" provided in the first byte is described as data for starting spinning of reels during a big bonus game. Data pertaining to the number of available big bonus games and data pertaining to the number of available big bonus operations are added to the second byte. The command code is transmitted when spinning of the reels 5a to 5c is started upon detection of actuation of the start switch 20 during a big bonus game. This command code is transmitted along with the game start command code during a normal game. A game start sound is generated as presentations. Further, presentations to be displayed during a big bonus game (three types of presentations for each big bonus stage) and the number of remaining normal games during a big bonus are effected. By means of the command code, a sub-sub command code (0Fh) for starting spinning of the reels during a big bonus is selected and transmitted.

Fig. 17 shows a data value corresponding to a command code (MCMD_RLSP) shown in Fig. 10, and a command code "07H" in the first byte is described as stop reel data. Stop reel information is added to the second byte. The command code is transmitted every time reels are stopped. Reel stop sound is generated, and sound effects are generated at the time of presentation of a chance. When any two of the reels 5a to 5c enter a *TEN P'AIS* state, chance presentation sound is generated in accordance with a sound solo presentation instruction to be described later. According to the type of a screen appearing on the liquid-crystal screen, another command code is selected and transmitted by means of the command code. Specifically, at the time of second stoppage of a reel, a reel stop sub-sub command (07h) (a *LI-ZHI* presentation instruction) for a normal reel screen is selected and transmitted. [*LI-ZHI* designates a game status analogous to *TEN P'AIS* status. This status arises when two identical symbols appear in predetermined positions, thereby inducing a chance of great winning. Throughout the specification, this status will be hereinafter referred to simply as "*LI-ZHI.*"] Further, at the time of first stoppage of a reel, a reel stop sub-sub command (08h) for an

internally-notified-state reel screen is selected according to the type of a screen appearing on the liquid-crystal screen, and the thus-selected sub-sub command is transmitted.

Fig. 18 shows a data value corresponding to a command code (MCMD_NHIT) shown in Fig. 10, and a command code "08H" in the first byte is described as winning data (exclusive of Jackpot winning data). Data pertaining to the type of winning, such as a big bonus, a regular bonus, or a replay winning, are added to the second byte. The command code is transmitted after all the reels 5a through 5c have stopped and the result of retrieval of winning has been acquired. In a case where reel lamps are blinked for presentation after all the reels 5a to 5c have stopped, the command code is transmitted after the presentation. Token payout sound and replay sound are generated as presentations. In the case of winning of bonus mode, bonus game presentation display is commenced. According to the state of the game at that time, another command code is selected and transmitted by means of the command code. When a normal reel screen is displayed, a small-jackpot winning presentation sub-sub command (09h) for a normal reel is selected and transmitted. When an internally-notified-state reel screen is displayed, small-jackpot winning presentation on an internally-notified-state reel screen is selected and transmitted. When a bonus winning has arisen, a bonus winning presentation sub-sub command (0Bh) is selected and transmitted. When a regular bonus winning has arisen during a big bonus game, a regular bonus winning presentation sub-sub command (11h) is selected and transmitted. In other cases; that is, when a winning has arisen during normal games provided in a big bonus game, a small-jackpot winning sub-sub command (010h) for a normal game during a big bonus is selected and transmitted.

Fig. 19 shows a data value corresponding to a command code (MCMD_JHIT) shown in Fig. 10. A command code "09H" stored in the first byte is described as jackpot (i.e., a bonus arising during the previously-described JAC game) winning data.

Data pertaining to the number of possible jackpot winnings (1 through 8 winnings) are added to the second byte. Data "0" to be added represent "losing." The command code is transmitted after all reels have stopped during a regular bonus game and a result of retrieval of winning has been produced. Jackpot winning sound is generated as a presentation, and display of a jackpot winning presentation and corresponding sound effects are effected (the sound effects are common to winning and losing presentations). By means of the command code, a jackpot winning presentation sub-sub command (0Eh) is selected and transmitted.

Fig. 20 shows a data value corresponding to a command code (MCMD_POFN) shown in Fig. 10. Command code "0AH" stored in the first byte is described as payout completion data. The command code is transmitted when payout is completed. Muting of payout sound is performed. When bonus winning has arisen, bonus start sound and bonus operation sound are generated.

Fig. 21 shows a data value corresponding to a command code (MCMD_BNST) shown in Fig. 10. Command code "0BH" stored in the first byte is described as bonus game status change instruction data. Data pertaining to the status of a big bonus and data pertaining to the status of a regular bonus are added to the second byte. The command code is transmitted at the end of a regular bonus game, at the end of a regular bonus game during a big bonus game, and at the end of the big bonus game. At the end of a regular bonus game, a regular bonus operation sound is muted. At the end of a regular bonus game played during a big bonus game, a regular bonus operation sound and a big bonus operation sound are muted. At the end of a big bonus game, muting of an operation sound corresponding to the status of a game (e.g., a game is ended normally or ended with punctures), display of a big bonus ending presentation, and generation of corresponding sound effects are effected. By means of the command code, a bonus stage display instruction sub-sub command (0Ch) is selected and transmitted.

Fig. 22 shows a data value corresponding to a command code

(MCMD_BBFN) shown in Fig. 10. Command code "0CH" stored in the first byte is described as data pertaining to an operation to be performed at the end of a big bonus game. Data pertaining to the status of a big bonus and data pertaining to the status of a regular bonus are added to the second byte. Status data pertaining to settlement operation and play-out operation are added to the second byte. In a case where the data stored in the second byte represent completion of a wait time after the end of a game (i.e., a continuation mode), the command code is transmitted after a game stop time has elapsed since the end of a big bonus game. In a case where the pay-out sound is being generated, the pay-out sound is muted. In a case where the data represent that a play-out sound is being generated, the sound is muted. In a case where the data stored in the second byte represent settlement operation, the command code is transmitted after the end of a big bonus game, and token payout sound is generated. In a case where the data stored in the second byte represent play-out operation, the command code is transmitted when the slot machine is set so as to stop a play at the end of a big bonus game, after checking of automatic settlement operation, or after completion of an operation. When pay-out sound is being generated, muting of the sound and generation of a play-out sound are performed.

Fig. 23 shows a data value corresponding to a command code (MCMD_ERR) shown in Fig. 10. Command code "0DH" stored in the first byte is described as error presentation data. Error status data are added to the second byte. The command code is transmitted upon detection of an error or upon recovery from an error. In the event of occurrence of an error, an error sound is generated. An error screen matching an error code is displayed. In contrast, at the time of recovery from an error, an error sound is muted, and the screen is restored to a status in effect prior to occurrence of the error. By means of this command code, an error screen sub-sub command (12h) is selected and displayed.

Fig. 24 shows a data value corresponding to a command code (MCMD_PSEL)

shown in Fig. 10. Command code "0EH" stored in the first byte is described as presentation type data. Data pertaining to the presentation type of the main control board 100 are added to the second byte. After the type of a presentation to be performed by the main CPU 101 has been selected upon detection of actuation of the start switch 20 during a normal game, the command code is transmitted along with the normal game start command described previously. A game start sound is generated as a presentation. At the time of presentation of a chance, corresponding sound effects are generated, and animated display of symbols on the liquid-crystal screen is commenced, thus effecting various chance presentations. By means of the command code, a sub-sub command is selected and transmitted according to the type of a reel screen. Specifically, in the case of a normal reel screen display, a reel spinning start sub-sub command (04h) for a normal reel screen is selected and transmitted. In the case of an internally-notified status reel screen, a spinning start sub-sub command (05h) for an internally-notified status reel screen is selected and transmitted.

Fig. 25 shows a data value corresponding to a command code (MCMD_SUND) shown in Fig. 10. Command code "0FH" stored in the first byte is described as sound single presentation instruction data. Data pertaining to details of a sound are added to the second byte. Here, the sound single presentation means presentation of a sound which enables determination of a specific presentation style through use of only information provided from the main control board 100, regardless of selection of a token insertion sound, a *LI-ZHI* sound originating from a main reel, and presentations processed by the sub-control board. More specifically, as will be described later, a *LI-ZHI* presentation is determined on the basis of data pertaining to an internally-generated combination or play information about the status of a game output from the main control board 100, as well as by reference to a presentation selection table stored in the sub-control board 200. As a matter of course, there is selected a sound presentation corresponding to

the type of a *LI-ZHI* presentation. In other words, a specific form of sound presentation cannot be determined from merely the play data output from the main control board 100. In contrast, sound presentation, such as token insertion sound, can be determined specifically, regardless of a presentation which is not specifically determined until selection is made by the sub-control board 200 (i.e., without involvement of a presentation selection table). The command code is transmitted when the main control board 100 has determined the type of sound described in the second byte. Accordingly, various sound effects are produced.

Timings within the stream of a game—at which commands are to be transmitted from the main control board 100—will be described by reference to Figs. 257 through 259.

In each of Figs. 257 through 259, the left-side portion of the drawing shows the schematic flow of a game, and the right-side portion of the same shows commands to be transmitted to the sub-control board 200 in respective processing steps pertaining to the game.

When the slot machine is powered up, operations of the CPU and RAM and those of other devices are checked, whereby the RAM and the I/O ports are initialized (M01_1). At this time, the presentation initialization command (01H) is transmitted to the sub-control board 200.

On the assumption that a game is to be played continuously, data pertaining to a previous game are cleared (M01_2). A determination is made as to whether or not a token automatic insertion request has been output at the time of commencement of a current game as a result of replay winning having arisen in the previous game (M01_3). If an insertion request has been issued, automatic token insertion processing is performed (M01_4). In contrast, if no automatic insertion request has been issued, a token is inserted by means of the player's token-inserting action or actuation of one of the bet switches 16 to 18 (M01_5). At this time, the gaming token insertion command (03H) and the sound presentation instruction

command (0FH) are transmitted. In the event of occurrence of an error (i.e., a token jam error or a token reverse-movement error), an error instruction command (0DH) is transmitted (M01_5).

Next, a determination is made as to whether or not the start switch 20 has been actuated (M01_6). When the start switch 20 has been actuated, a determination is made as to whether or not a time of 4.1 seconds has elapsed from the previous game (M01_7). If a time of 4.1 seconds has not yet elapsed, commencement of a game is awaited until a specific period of time elapses (M01_8). Further, there is a necessity for generating a wait sound, and hence a sound presentation instruction command (0FH) is transmitted.

A random number for determining winning is extracted (M01_9), and a watchdog timer is set for ensuring a specific period of time for one play (M01_10). Probability sampling is performed through use of the thus-extracted random number and a probability table (M01_11). By reference to the result of probability sampling and the status of a game, selection is made as to the type of a game start sound, to the type of a reel lamp demonstration, and to a presentation for which the main control board 100 determines whether to illuminate WIN lamps in the case of bonus winning (M01_12). At this time, according to the type of the game, the presentation type command (0EH) is transmitted from the main CPU 101 along with any one of the command for starting a game during a normal game (04H), the command for starting spinning of reels during a regular bonus game (05H), and the command for starting spinning of reels during a big bonus game (06H). A determination is made as to whether or not the player has performed a stop operation after the stop switches 21a through 21c have been made active (M01_13).

Here, if the stop switches 21a through 21c have not yet been actuated, a determination is made as to whether or not an automatic reel stop time (usually a period of 30 to 40 seconds since reels have started to spin) has been reached. If the time has not yet been reached, processing is looped to step "M01_13" (M01_14).

In the event that a stop request has been issued, the number of frames over which the reels are to coast before stopping from the positions of the currently-displayed symbols (the frames will be hereinafter called simply "coasting frames") is determined, by reference to a stop table selected by means of a generation flag or a stop operation position (M01_15). After having spun over the coasting frames, the reels 5a through 5c are stopped (M01_16). At this time, the reel stop command (07H) and the sound presentation instruction command (0FH) for generating a reel stopping sound are transmitted.

A determination is made as to whether or not all the reels 5a through 5c have stopped. If not all of the reels 5a through 5c have stopped yet, processing loops to step "M01_13" (M01_17). In contrast, if all the reels 5a through 5c have already stopped, the reel lamp blinking operation (M01_18) and the WIN lamp illumination operation (M01_19), which are to be performed after all reels have stopped, are carried out in accordance with a selected report pattern (M01_19).

Next, winning is retrieved (M01_20), and a determination is made as to whether or not a mismatch exists between the pattern of the reels 5a through 5c after stopping and an internally-generated combination (M01_21). If a mismatch has arisen, occurrence of an illegal error is reported, and an error instruction command (0DH) is transmitted (M01_22). If no mismatch is found, token credits or tokens are paid out (M01_23). At this time, the jackpot winning command (09H) or the winning command (08H) exclusive of a jackpot is transmitted according to the state of the game. After the end of payout of tokens, a payout completion command (0AH) is transmitted. In the event of occurrence of an error, such as an empty hopper or a hopper jam, the error instruction command (0DH) is transmitted.

A determination is made as to whether or not a big bonus game or a regular bonus game is in progress (M01_24). If neither a big bonus game nor a regular bonus game is being played, processing returns to the start. In contrast, if a big bonus game or a regular bonus game is being played, a check is made as to the

progress of the bonus game (M01_25). The bonus game status change instruction command (0BH) matching the status of gaming is transmitted.

A determination is made as to whether or not the big bonus game has ended (M01_26). If the big bonus game has ended, initialization operation to be performed at the end of a big bonus game (i.e., clearing of RAM) is effected (M01_27), and processing returns to the start.

<Commands to be Transmitted from the Sub-Control Board to the Image Control Board>

Display control codes to be transmitted to the image control board 300 from the liquid crystal display command transmission ports (shown in Fig. 8) are shown in Fig. 26. Fig. 26 shows display control codes transmitted from the liquid crystal display command transmission ports. Command codes are provided in the left-side column; data values are provided in the center column; and details of the commands are described in the right-side column.

Data pertaining to each of the command codes consist of one byte (= eight bits). Data pertaining to each command code will be described in more detail by reference to Figs. 27 through 45.

Figs. 27 through 45 show data values corresponding to typical command codes. Contents of respective bytes are described in descending sequence from the top of the code.

Fig. 27 shows a data value corresponding to a command code (DSP_INIT) shown in Fig. 26. Command code "01h" is described as liquid crystal display erasure (initialization) data. The command data are transmitted at the time of initialization operation or when a presentation initialization command is received. The command data correspond to the command data shown in Fig. 11. The command is transmitted at the time of an initialization operation or when a presentation initialization command is received from the main control board 100.

Fig. 28 shows a data value corresponding to a command code (DSP_DEMO)

shown in Fig. 26. Command code "02h" is described as demonstration display data. These command data are transmitted when demonstration display is performed. The command data correspond to the command data shown in Fig. 12. The command code is transmitted when the demonstration display command is received.

Fig. 29 shows a data value corresponding to a command code (DSP_REEL) shown in Fig. 26. Command code "03h" is described as reel screen display data. Reel screen type data are added to this command code. When a gaming token insert command is received, the command code is transmitted along with the displayed symbol data preserved in the sub-control board 200. During the course of play of a bonus game, the command code is not transmitted. The command code corresponds to the command data shown in Fig. 13 and is transmitted when the sub-control board 200 receives a demonstration display command from the main-control board 100.

The displayed symbol data comprise displayed symbol data 1 pertaining to a left-side displayed symbol, and displayed symbol data 2 pertaining to a center displayed symbol and a right-side displayed symbol. As shown in Fig. 30, each of displayed data sets comprises ten types of symbols; i.e., "7," "BAR," "Do," "Cake," "Cookie," "E," "X," "T," "R," and "A." So long as four bits are assigned to the left-side displayed symbol, the center displayed symbol, and the right-side displayed symbol, the type of symbol can be identified.

Fig. 31 shows a data value corresponding to a command code (DSP_NSTR) shown in Fig. 26. Command code "04h" is described as data for starting spinning of reels on a normal reel screen. Sign presentation instruction data are added to this command code. The sub-control board 200 transmits the command data by means of selecting a sign presentation, upon receipt of a presentation type command from the main control board 100. The command data are based on the command data shown in Fig. 24. In connection with the command data, selection of stationarily-displayed symbols and selection of *LI-ZHI* presentation type are

performed simultaneously. Transmission of the command data pertaining to the symbols and transmission of the presentation type are performed by means of different commands. When a command pertaining to the type of a presentation to be performed by the main CPU 101 is received, a presentation is selected, and then the command data are transmitted.

Fig. 32 shows a data value corresponding to a command code (DSP_SSTR) shown in Fig. 26. Command code "05h" is described as data for starting spinning of reels on an internally-notified-status reel screen. Sign presentation instruction data are added to this command code.

In a case where spinning of the reels 5a through 5c is started on the internally-notified-status reel screen, there is no necessity for performing presentation of internal generation of a bonus combination (i.e., a *LI-ZHI* presentation), because determination of a bonus combination has already been displayed. For this reason, a *LI-ZHI* presentation sign is not included in the transmission command.

When the internally-notified-status reel screen is displayed and a presentation type command is received from the control board 100, selection of a presentation is effected, and the command data are transmitted. The command data correspond to those shown in Fig. 24. In connection with the command data, selection of stationarily-displayed symbols is performed. Transmission of the command data pertaining to the stationarily-displayed symbols is performed by means of another command. Here, the internally-notified status means a status in which internal generation of a bonus is definitely displayed on the liquid-crystal screen on the basis of the determination data determined by the main control board 100. When a command pertaining to the type of presentation to be performed by the main CPU 101 is received, selection of a presentation is made, and the command data are transmitted. Although selection of stationarily-displayed symbols and selection of type of *LI-ZHI* presentation are performed simultaneously, command

data pertaining to them are transmitted by different commands.

Fig. 33 shows a data value corresponding to a command code (DSP_NLSP) shown in Fig. 26. In the present embodiment, command code "06h" is unassigned.

Fig. 34 shows a data value corresponding to a command code (DSP_RECH) shown in Fig. 26. Command code "07h" is described as data (*LI-ZHI* presentation instruction) for stopping spinning of reels on a normal reel screen. *LI-ZHI* presentation instruction data are added to this command code. In a case where a stop command is received from the main control board 100 and the stop command induces first stoppage and represents a normal reel screen (i.e., during the course of a normal game), the command data are transmitted. Hence, the selected *LI-ZHI* presentation instruction and stationarily-displayed symbol data are transmitted. As shown in a lower column in Fig. 34, the stationarily-displayed symbol data comprise a stop sequence signal, stationarily-displayed-symbol data 1, and stationarily-displayed-symbol data 2.

In the present embodiment, in a case where an instruction for advance presentation concerning determination of a big or regular bonus is issued through use of a reel-spinning-start command, symbols to be displayed on the liquid-crystal screen are automatically stopped. In such a case, the command is not transmitted. The command data correspond to those shown in Fig. 18.

Fig. 35 shows a data value corresponding to a command code (DSP_SSTP) shown in Fig. 26. Command code "08h" is described as reel stop data for an internally-notified-state reel screen. The command code is transmitted along with the stationarily-displayed-symbol data. As shown in a lower column in Fig. 35, the stationarily-displayed symbol data comprise a stop sequence signal, stationarily-displayed-symbol data 1, and stationarily-displayed-symbol data 2. The command data correspond to those shown in Fig. 17.

Fig. 36 shows a data value corresponding to a command code (DSP_NHIT) shown in Fig. 26. Command code "09h" is described as small-jackpot winning

presentation data for a normal reel screen. Small-jackpot winning presentation instruction data are added to this command code. When a winning command is received while a normal reel screen is displayed, the command data are transmitted along with the stationarily-displayed-symbol data. As shown in a lower column in Fig. 36, the stationarily-displayed symbol data comprise stationarily-displayed-symbol data 1 and stationarily-displayed-symbol data 2. The command data correspond to those shown in Fig. 18.

The stationarily-displayed-symbol data to be transmitted are identical with those transmitted by the reel stop command.

Fig. 37 shows a data value corresponding to a command code (DSP_SHIT) shown in Fig. 26. Command code "0Ah" is described as small-jackpot winning presentation data for an internally-notified-status reel screen. Small-jackpot winning presentation instruction data are added to this command code. When a winning command is received from the main control board 100 while an internally-notified-status reel screen is displayed, the command data are transmitted along with the stationarily-displayed-symbol data. As shown in a lower column in Fig. 37, the stationarily-displayed symbol data comprise stationarily-displayed-symbol data 1 and stationarily-displayed-symbol data 2. The command data correspond to those shown in Fig. 18.

The stationarily-displayed-symbol data to be transmitted are identical with those transmitted by the reel stop command.

Fig. 38 shows a data value corresponding to a command code (DSP_BHIT) shown in Fig. 26. Command code "0Bh" is described as bonus winning presentation data. Type-of-bonus-won data are added to this command code. When a winning command is received from the main control board 100 and where a bonus winning is instructed, the command data are transmitted. The command data correspond to those shown in Fig. 18.

In a case where a reel spinning start command is received from the main

control board 100 before completion of a bonus winning presentation, a display command is transmitted in accordance with the command. In this case, the bonus winning presentation is forcefully terminated.

Fig. 39 shows a data value corresponding to a command code (DSP_BSTG) shown in Fig. 26. Command code "0Ch" is described as bonus stage display data. Bonus stage type data are added to this command code. When a bonus-game-status change instruction command is received from the main control board 100, the command data are transmitted. The command data correspond to those shown in Fig. 21.

Fig. 40 shows a data value corresponding to a command code (DSP_RSTR) shown in Fig. 26. Command code "0Dh" is described as data for starting spinning of reels during a regular bonus game. Data pertaining to the number of regular bonus games and data pertaining to the number of big bonus games are added to this command code. When a command for starting spinning of reels during a regular bonus game is received from the main control board 100, the command data are transmitted. The command data correspond to those shown in Fig. 15.

Fig. 41 shows a data value corresponding to a command code (DSP_JHIT) shown in Fig. 26. Command code "0Eh" is described as jackpot winning presentation data. Data pertaining to the number of regular bonus stages and data pertaining to the number of jackpot wins are added to this command code. When a command for starting spinning of reels during a regular bonus game is received from the main control board 100, the command data are transmitted. The command data correspond to those shown in Fig. 19.

Fig. 42 shows a data value corresponding to a command code (DSP_BSTR) shown in Fig. 26. Command code "0Fh" is described as data for starting winning of reels during a big bonus. Data pertaining to the number of big bonus stages and data pertaining to the number of big bonus games are added to this command code. When a command for starting spinning of reels during a big bonus game is received

from the main control board 100, the command data are transmitted. The command data correspond to those shown in Fig. 16.

Fig. 43 shows a data value corresponding to a command code (DSP_BNHT) shown in Fig. 26. Command code "10h" is described as data pertaining to small-jackpot winning presentation during a big bonus game. Data pertaining to the number of big bonus stages, data pertaining to the number of remaining big bonus games, and small-jackpot winning presentation instruction data are added to this command code. When a winning command is received from the main control board 100 during the big bonus game and the thus-received winning command is other than a regular bonus, the command data are transmitted. As shown in a lower column in Fig. 36, the stationarily-displayed symbol data comprise stationarily-displayed-symbol data 1 and stationarily-displayed-symbol data 2. The command data correspond to those shown in Fig. 18.

Fig. 44 shows a data value corresponding to a command code (DSP_BRHT) shown in Fig. 26. Command code "11h" is described as presentation data pertaining to winning of a regular bonus during a big bonus game. Data pertaining to the number of regular bonus stages are added to this command code. When a winning command is received from the main control board 100 during the big bonus game and the thus-received winning command corresponds to a regular bonus, the command data are transmitted. The command data correspond to those shown in Fig. 18.

Fig. 45 shows a data value corresponding to a command code (DSP_ERR) shown in Fig. 26. Command code "12h" is described as error screen display data. Error type data are added to this command code. When an error presentation instruction command is received from the main control board 100, the command data are transmitted. The command data correspond to those shown in Fig. 23.

In a case where an error status cancel instruction command is received from the main control board 100, the display command which has been saved is

transmitted, thereby restoring a display status before occurrence of an error.

<Constants to be Used in the Control Processing Performed by the Sub-Control Board>

A plurality of constants to be used in control processing are defined in the sequence program to be executed by the sub-control board 200.

Fig. 46 shows constants to be used in the control processing performed by the sub-control board 200. The left-side column shows labels; the center column shows data values; and the right-side column shows descriptions of the command codes.

For instance, "DUMMY" denotes dummy data and assumes an initial value of 0. "STBI_BITN" denotes a detection bit included in the strobe signal output from the main CPU 101 and assumes an initial value of 7.

Each of the constants is initialized to the corresponding value shown in Fig. 46 when a reset signal is received from the main control board 100.

<Timer to be Used in the Control Processing Performed by the Sub-Control Board>

A plurality of timers to be used in control processing are defined in the sequence program to be executed by the sub-control board 200.

Fig. 47 shows timers to be used in the control processing performed by the sub-control board 200. The left-side column shows labels; and the right-side column shows descriptions of the timers and timeout values.

"MDWIN_TM" denotes a time to detect down-status of the main CPU. A timeout is set to 50 ms.

Respective timers are used in the control processing to be performed by the sub-control board 200.

<Flags to be Used in the Control Processing Performed by the Sub-Control Board>

Fig. 48 shows the detailed internal construction of an especially-dominant

work area for selecting a presentation during the control processing performed by the sub-control board 200. The left-side column shows labels; and the right-side column shows descriptions of the flags.

For example, "GAMEST; [D7]" denotes a flag to be set when a big bonus game is in progress, and "GAMEST; [D6]" denotes a flag to be set when internal generation of winning is being notified.

The flags are used for the control processing in the sub-control board 200.

<Work Area>

A plurality of work areas to be used in the control processing are defined in the sequence program to be executed by the sub-control board 200.

Figs. 49 and 50 show work areas to be used in the control processing to be performed by the sub-control board 200. The left-side column shows labels; the center column shows the number of bytes of an employed location; and the right-side column shows descriptions of the work areas.

For instance, "GAMEST" denotes a one-byte work area to be used in connection with the status of a game, and "PRDC_STS" denotes a one-byte work area to be used in connection with a presentation-status flag.

The work areas are used in the control processing performed by the sub-control board 200.

<Sequence Control Table>

A plurality of sequence control tables to be used in sound-effect generation control processing are stored in the program ROM 202 of sub-control board 200, and a sequence control table corresponding to a selected presentation is selected. Fig. 51 shows a sequence control table stored in the program ROM 202 of the sub-control board 200. The table is constituted of a plurality of sequence control tables corresponding to *LI-ZHI* statuses and game statuses, such as a big bonus game and a regular bonus game.

For instance, there are sequence control tables corresponding to a balloon

LI-ZHI (a presentation for determining a regular bonus) and a balloon *LI-ZHI* (a presentation for determining a big bonus), and a sound generation sequence corresponding to the status of a game is described in each of the sequence control tables.

The sound-source IC 206 operates on the basis of the respective sequence control tables and generates sound effects from the speaker 25 by way of the power amplifier 207.

<Commands to be Transmitted to the Sound IC>

As shown in Fig. 52, command data to be transmitted to the sound-source IC 206 consist of four bytes (one byte = eight bits).

Data pertaining to the type of sound, such as an alarm sound, and data pertaining to a channel used are stored in the first byte of the command data, and replay-level data are stored in the second byte. A pan-pot setting is stored in the third byte, and phrase-number data are stored in the fourth byte.

<Sound Output Request Control Code>

The previously-described sound effects are generated by means of a sound output request control code.

Figs. 58 through 60 show sound output request control codes; in each Fig. the left-side column shows labels; the center column shows the types of sound; and the right-side column shows descriptions of the control codes.

For instance, sound output request code "NONSD" assumes a sound type of "0" and makes a request for generating sound from presentation sequence control data (no sound output), and sound output request code "ENDSQ" assumes a sound type of "OFFH" and makes a request for generating sound from presentation sequence control data (end of a sequence).

<Sound Output>

Figs. 53 through 57 show specific sound effects to be generated by the sound-source IC 206.

In a case where there is a necessity of generating sound effects, such as presentation sounds like *LI-ZHI* presentation sounds or *bonus-in-progress* presentation sounds, over a comparatively long period of time, a single group of sound effects are produced by combination of the sound output request control codes, so that a sound-effect sequence control operation is effected, to thereby produce sounds. More specifically, sound effects are generated in accordance with the sequence control table shown in Fig. 51.

For instance, in a case where the sub-control board selects "PowerBall 3 *LI-ZHI* Losing" in accordance with a command output from the main control board 100, a *LI-ZHI* presentation instruction command is transmitted to the image control board 300. With regard to an image, the image control CPU 301 analyzes the thus-received command and controls the image control IC 304, on the basis of the image sequence control table stored in the program ROM 302 mounted on the image control board 300, to thereby sequentially display *LI-ZHI* presentation scenes. Similarly, the sub-CPU 201 selects "PowerBall 3 *LI-ZHI* Losing" from the sound-effect sequence control table stored in the program ROM 202, thereby performing presentation processing while ensuring a match between image presentations and time.

By reference to Fig. 53, there will now be described sound effects which are produced by the sound-source IC 206 when the sequence control table for "PowerBall 3 *LI-ZHI* Losing" is selected. Here, "PowerBall" and "Do!" denote characters to be displayed on the image display section 13.

When the sequence control table for "PowerBall 3 *LI-ZHI* Losing" is selected, the sound-source IC 206 sequentially generates sounds at specified times: for example, no sound for 650 ms; a super *LI-ZHI* advancement sound for 167 ms; a "Do!" emergence sound for 1100 ms; a powerball straining sound (sound generated when a character strains to accumulate power) for 1683 ms; a powerball ascending sound for 933 ms; a powerball ascending

sound for 367 ms; a muted powerball-straining sound for 700 ms; a target-lock sound for 933 ms; a powerball throwing sound for 350 ms; a powerball explosion sound for 150 ms; a powerball explosion sound for 167 ms; a powerball explosion sound for 1167 ms; and a sitting-down sound. Upon detection of an end code, the sound-source IC 206 terminates the presentation sequence control operation. The thus-generated sound effects are produced from the speaker 25 by way of the power amplifier 207.

Similarly, when the sequence control table for "PowerBall 3 *LI-ZHI* Winning" is selected, sound effects such as those shown in Fig. 54 are produced. When a sequence control table for "Regular-Bonus Stage 3 Ended with Eight Wins in Jackpots" is selected, sound effects such as those shown in Fig. 55 are generated. When a sequence control table for "Regular-Bonus Stage 3, Jackpot Losing Ended with Punctures" is selected, sound effects such as those shown in Fig. 56 are generated. When a sequence control table for "Regular-Bonus Stage 3, Jackpot Winning Ended with Punctures" is selected, sound effects such as those shown in Fig. 57 are generated.

<Sound Output Data Table>

When any of the sound output request control codes are transmitted, the sound-source IC 206 generates sound effects in accordance with the sound output data table.

Figs. 61 through 75 show sound output tables; in each Fig. the left-side column shows code names assigned to sound output data; the center column shows sound output data; and the right-side column shows descriptions of the sound output data.

For instance, when a sound output request control code (SD_ERR) is transmitted, a sound output data table relating to an anomalous operation alarm sound (indicated by 6 in Fig. 61) is selected. In the sound output data table, ERR_SW indicates that the type of sound is an alarm sound; REP indicates that the

sound is continuous; CH1 indicates that channel 1 is used for generating the alarm sound; 20 indicates a level of sound; CENTER indicates that a pan-pot is set to center; and 1 indicates a phrase number.

<Sound Code>

Next will be described sound codes used in the sound output data tables.

Fig. 76 shows sound codes used in the sound output data tables, wherein the left-side column shows labels; the center column shows data values; and the right-side column shows descriptions of the sound codes.

For instance, "INIT_CODE" denotes an initialization request code and assumes a data value of "0E0H," and "CMD_PLAY" denotes a replay start command and assumes a data value of "0F0H."

<*LI-ZHI* Presentation Selection Table>

Various presentations corresponding to gaming statues are performed in the image display section 13. Presentations to be performed in the image display section 13 are associated with generation of a *LI-ZHI* and are divided into a *LI-ZHI* sign presentation and a *LI-ZHI* presentation. There will be described a *LI-ZHI* presentation selection data table used in association with generation of a *LI-ZHI*, by reference to Figs. 77 through 91.

Figs. 77 through 81 show *LI-ZHI* presentation selection tables to be used during a normal game; Figs. 82 through 86 show *LI-ZHI* presentation selection tables to be used during internal generation of a bonus mode; and Figs. 87 through 91 show *LI-ZHI* presentation selection tables to be used during illumination of WIN lamps.

In each of Figs. 77 through 91, the left-side column shows the type of *LI-ZHI* sign presentation; the center column shows the type of *LI-ZHI* presentation; and the right-side column shows reference values to be used for selecting a *LI-ZHI* presentation. The reference values (a total of 20 types ranging from 00 to 19) to be used for selecting a *LI-ZHI* presentation are categorized according to a plurality of

flash data sets. A flash data set to be used for categorizing the reference values is selected according to a selection table shown in Fig. 98.

In a table shown in Fig. 98 for selecting a *LI-ZHI* selection table, a reference value is selected for each of the *LI-ZHI* selection tables shown in Figs. 77 through 91, on the basis of the presentation data output from the main control board 100, the status of gaming and a starting sound (on the longitudinal axis), and a flash data number (on the lateral axis). A *LI-ZHI* presentation selection table is selected in accordance with the state of gaming, and a *LI-ZHI* presentation and a *LI-ZHI* sign presentation are determined from a random number for selecting a presentation and from a reference value. Flash table number "0" corresponds to NO FLASH DATA described in the table shown in Fig. 98 to be described later. Random numbers to be used for sampling consist of 65,536 (from "0" to "65535"). Further, *LI-ZHI* presentation selection (*LI-ZHI* sign presentation selection) tables are roughly classified into a selection table for use in the course of a normal game shown in Figs. 77 through 81 and a selection table for use in the course of an internally-generated bonus game with illumination of WIN lamps (indicating that determination data are selected). By reference to a corresponding column of reference values, the random number is compared with reference values in descending sequence from the top (each of the selection tables is split into five or six segments for convenience of illustration; in reality, each selection table is continuous).

In the case of a *LI-ZHI* presentation selection table for use in the course of a normal game shown in Fig. 77, if a reference value selected this time assumes a value of "00" and a random number is sampled from the numbers "0" to "58732," "no *LI-ZHI* presentation" is selected. When a random number is sampled from the numbers "58733" to "61232," "normal *LI-ZHI* losing presentation" is selected. When a random number is sampled from the numbers "61233" to "63232", "*oshikura* losing presentation" is selected." When a random number is sampled from the numbers "63233" to "64232," "powerball 1 *LI-ZHI* losing presentation" is selected.

When a random number is sampled from the numbers "64233" to "64532," "powerball 2 *LI-ZHI* losing presentation" is selected. When a random number "64533" is sampled, "powerball 3 *LI-ZHI* losing presentation" is selected. When a random number is sampled from the numbers "64534" to "65333," "balancing-on-rolling-ball right *LI-ZHI* losing presentation" is selected. When a random number is sampled from the numbers "65335" to "65533," "balancing-on-rolling-ball left *LI-ZHI* losing presentation" is selected. When a random number "65534" is sampled, "balancing-on-rolling-ball center *LI-ZHI* losing presentation" is selected. No *LI-ZHI* sign presentations are provided for these selections. When a random number "65535" is sampled, processing proceeds to the selection table shown in Fig. 78 (as mentioned previously, the tables shown in Figs. 77 through 81 constitute a single continuous selection table in accordance with figure numbers), wherein a *OSHIKURA LI-ZHI* losing presentation is selected (*OSHIKURA* comes from the name of a popular Japanese wintertime game, in which children push each other to warm up in the cold; and *OSHIKURA* is used as is throughout the specification). Accordingly, an *OSHIKURA LI-ZHI* presentation is selected as a *LI-ZHI* sign presentation.

A specific computation method will be described. The sampled random number is subtracted from reference number "58732" on the first row, to thereby determine a result of first computation. If the result of first computation represents a value of 0 or greater, "No *LI-ZHI* presentation" on the first row is selected. If the result of first computation is negative, the absolute value of the result of first computation is subtracted from second reference numeral "2500" (here, the reference value differs from that selected by means of the flash data), to thereby determine a result of second computation. If the result of second computation is greater than 0, "normal *LI-ZHI* losing presentation" on the second row is selected. If the result of second computation is negative, the absolute value of the result of second computation is subtracted from third reference numeral "2000," to thereby

determine a result of third computation. If the result of third computation is greater than 0, "*Oshikura LI-ZHI* losing presentation" on the third row is selected. Computations are performed sequentially in the same manner, and *LI-ZHI* presentations are selected so as to correspond to random numbers up to "65535."

<Flash Data Selection Table to be Used in *LI-ZHI* Presentation Selection Table>

A flash data selection table to be used in the previously-described *LI-ZHI* presentation selection table will be described by reference to Fig. 98.

As mentioned above, in a case where a *LI-ZHI* presentation is selected by reference to the *LI-ZHI* presentation selection table, a reference value is utilized. More specifically, reference values on the *LI-ZHI* presentation selection table in the lateral direction are determined from blinking patterns (flash data) 0 through 8 to be performed after stoppage of all reels, the patterns being transmitted from the main control board 100, game start sounds 1 and 2, and the status of gaming (e.g., a normal game is in progress, an internally-generated bonus game is in progress, or an internally-generated bonus game is in progress with WIN lamps illuminated).

<Symbols to be Displayed when a *LI-ZHI* Presentation is effected>

Symbols to be displayed during a *LI-ZHI* presentation will be described by reference to Fig. 92 through 95.

Fig. 92 shows a BBRECHDAT_BB table for selecting symbols displayed when a big bonus generation presentation is effected; Fig. 93 shows a RBRECHDAT_RB table for selecting symbols displayed when a regular bonus generation presentation is effected; Fig. 94 shows a MSRECHDAT table for selecting symbols displayed when a *LI-ZHI* losing presentation is effected; and Fig. 95 shows a table for selecting a center symbol displayed when a "balancing-on-rolling-ball *LI-ZHI* losing presentation" is effected.

In each of Figs. 92 through 94, the left-side column shows the type of displayed symbols, and the right-side column shows descriptions of sampling

numbers. In Fig. 95, the left-side column shows *TEN P'AIS* symbols, and the right-side column shows descriptions of the center displayed symbols.

Procedures for selecting a displayed symbol will be described by means of taking, as an example, a "'BBRECHDAT_BB' *LI-ZHI* presentation displayed-symbol table." In the case of selection of symbols to be displayed at the time of a *LI-ZHI* presentation, symbols to be displayed are determined by means of comparison between a sampled random number and each of the reference values described in rows. 65536 numbers (from "0" to "65535") are employed as random numbers for sampling.

For instance, in connection with the "BBRECHDAT_BB" *LI-ZHI* presentation displayed-symbol table, when a random number falling within the range of "0" to "19660" is sampled, displayed symbol "7" is selected. When a random number falling within the range of "19661" to "45874" is sampled, displayed symbol "Do" is selected. When a random number falling within the range of "45875" to "55705" is sampled, displayed symbol "Cake" is selected. When a random number falling within the range of "55706" to "65535" is sampled, displayed symbol "Cookie" is selected.

A specific computation method will be described. The sampled random number is subtracted from reference number "19660" on the first row, to thereby determine a result of first computation. If the result of first computation represents a value of 0 or greater, "7" on the first row is selected. If the result of first computation is negative, the absolute value of the result of first computation is subtracted from second reference numeral "26214," to thereby determine a result of second computation. If the result of second computation is greater than 0, "Do" on the second row is selected. If the result of second computation is negative, the absolute value of the result of second computation is subtracted from third reference numeral "9831," to thereby determine a result of third computation. If the result of third computation is greater than 0, "Cake" on the third row is selected.

Computations are performed sequentially in the same manner, and symbols to be displayed are selected so as to correspond to random numbers up to "65535."

When a "balancing-on-rolling-ball *LI-ZHI* losing presentation" is effected, a center displayed symbol to be described in the right-side column is selected so as to correspond to the *TEN P'AIS* symbol described in the left-side column. For example, in the case of *TEN P'AIS* symbols, in which the left-side displayed symbol is "7" and the right-side displayed symbol is "7," "BAR" is displayed in the center.

<Displayed Symbol Selection Table at the time of No *LI-ZHI*>

The type of a winning sign presentation and displayed symbols, which are used at the time of no *LI-ZHI* presentation, will be described by reference to Figs. 96 and 97. The type of a winning sign presentation to be described herein in principle represents the role of notifying internal generation of a small-jackpot combination. Notification of internal generation of a bonus combination is not effected actively; internal generation of a bonus combination is notified by means of a *LI-ZHI* presentation.

Fig. 96 shows a table for selecting symbols to be displayed during a normal game (without a *LI-ZHI*), and Fig. 97 shows a table for selecting symbols to be displayed during internal generation of a bonus combination (without a *LI-ZHI*).

As shown in Fig. 96, in the case of a normal game (without a *LI-ZHI*), a plurality of displayed-symbol selection tables are prepared so as to correspond to internally-generated combinations. In each of the displayed-symbol selection tables, correspondence exists between a sampled random number, the type of a winning sign presentation, and the type of a displayed symbol.

When "Cherry or Dragon 2" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 1" is selected. When "Dragon 1" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 2" is selected. When "Diamond" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 3" is selected. When "Replay" is

selected as an internally-generated combination, a "Displayed Symbol Selection Table 4" is selected. When "Single-Shot RB" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 9" is selected. When "Single-Shot BB" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 10" is selected. When "Losing" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 0" is selected. Here, the term "Single-Shot RB" signifies a game in which a regular bonus game is internally generated. Further, the term "Single-Shot BB" signifies a game in which a big bonus game is internally generated. During the course of a normal game, a match exists between an internally-generated combination and a winning sign presentation.

As shown in Fig. 97, in a case where a bonus game is being internally generated (without a *LI-ZHI*), a plurality of displayed symbol selection tables are prepared so as to correspond to an internally-generated combination. For this reason, a mismatch often arises between a winning sign presentation and a real internally-generated combination during internal generation of a bonus. For instance, if winning has not arisen in a replay game even though a replay sign presentation has been performed (as a matter of course, a *MEOSHI* operation has been performed), a bonus is determined (*MEOSHI* is a technique for stopping a spinning reel such that an intended symbol is displayed). In each of the displayed-symbol selection tables, correspondence exists between a sampled random number, the type of a winning sign presentation, and the type of a displayed symbol.

When "Group 1" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 5" is selected. When "DG (Dragon)" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 6" is selected. When "Diamond" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 7" is selected. When "Replay" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 8" is

selected. When "RB" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 11" is selected. When "BB" is selected as an internally-generated combination, a "Displayed Symbol Selection Table 12" is selected.

Procedures for selecting the type of a winning sign presentation and a displayed symbol will be described by means of taking, as an example, a "Displayed Symbol Selection Table 1." In the case of selection of type of a winning sign presentation and a displayed symbol at the time of no *LI-ZHI* presentation, the type of a winning sign presentation and a symbol to be displayed are determined by means of comparison between a sampled random number and each of the reference values described in rows. 65536 numbers (from "0" to "65535") are employed as random numbers for sampling.

For instance, in connection with the "Displayed Symbol Selection Table 1," when a random number falling within the range of "0" to "52428" is sampled, "Group 1L Sign Presentation" is selected. When a random number falling within the range of "52429" to "58982" is sampled, "Group 1H Sign Presentation" is selected. Further, when a random number falling within the range of "58983" to "65535" is sampled, no winning sign presentation is effected.

A specific computation method will be described. The sampled random number is subtracted from reference number "52428" on the first row, to thereby determine a result of first computation. If the result of first computation represents a value of 0 or greater, "Group 1L Sign Presentation" on the first row is selected. If the result of first computation is negative, the absolute value of the result of first computation is subtracted from second reference numeral "6554," to thereby determine a result of second computation. If the result of second computation is greater than 0, "Group 1H Sign Presentation" on the second row is selected. If the result of second computation is negative, "None" on the third row is selected.

The type of a displayed symbol will be selected by means of the same computation method. The sampled random number is subtracted from reference number "3276" on the first row, to thereby determine a result of first computation. If the result of first computation represents a value of 0 or greater, "X" on the first row is selected. If the result of first computation is negative, the absolute value of the result of first computation is subtracted from second reference numeral "45874," to thereby determine a result of second computation. If the result of second computation is greater than 0, "T" on the second row is selected. If the result of second computation is negative, the absolute value of the result of second computation is subtracted from third reference numeral "3276," to thereby determine a result of third computation. If the result of third computation is greater than 0, "R" on the third row is selected. Computations are performed sequentially in the same manner, and symbols to be displayed are selected so as to correspond to random numbers up to "65535."

<Flash Data Table>

The relationship between flash data, a random number for selection purpose, a reel blinking pattern, and presentation sound (start sound) will now be described by reference to Figs. 99 through 102.

As shown in Figs. 99 through 102, flash data tables are of ten types ("0" to "9"). The flash data tables correspond to respective game statuses. Flash data table "9" corresponds to a winning mode of "Diamond" arising during internal generation of a big bonus or a regular bonus. Flash data table "0" corresponds to a losing mode arising in a normal game when a big bonus game is inoperative. Flash data table "1" corresponds to a group 1 winning mode during a normal game (generation of a group, two cherries, four cherries, and dragon B) when a big bonus game is inoperative. Flash data table "2" corresponds to a winning mode of "Dragon" during a normal game when a big bonus game is inoperative. Flash data table "3" corresponds to a winning mode of "Diamond" during a normal game when a

big bonus game is inoperative. Flash data table "4" corresponds to a winning mode of "Replay" during a normal game when a big bonus game is inoperative. Flash data table "5" corresponds to a winning mode of a big bonus or a regular bonus during a normal game when a big bonus game is inoperative. Flash data table "6" corresponds to losing during a big bonus or a regular bonus. Flash data table "7" corresponds to a group 1 winning mode during internal generation of a big bonus or a regular bonus. Flash data table "8" corresponds to a winning mode of "Dragon" or "Replay" during internal generation of a big bonus or a regular bonus.

In each of the flash data tables, a combination of a random number and winning determination data corresponds to any one of the two types of game start sounds; i.e., "Start Sound 1" and "Start Sound 2," and any one selected from nine types of reel blinking patterns (patterns "1" to "8" and "no blinking").

As shown in Fig. 99, in the case of random number "28" and winning determination data "none" described in the flash data table "9," "Start Sound 1" and reel blinking pattern "4" are selected. In the case of random number "225" and winning determination data "none" described in the flash data table "0," "Start Sound 1" and reel blinking pattern "None" are selected.

The reel blinking pattern means a pattern in which nine back lamps 125 in total, three back lamps being arranged in a vertical row in each of the three reels 5a to 5c, are illuminated or extinguished or blinked. As shown in Figs. 261 to 268, reel blinking pattern "1" involves extinction of all back lamps 125 for a period of 103.25 ms (see Fig. 261). Reel blinking pattern "2" involves continuous repetition of eleven types of illumination patterns within a travel time of 150.18 ms, to thereby cause blinking of the back lamps 125 (see Fig. 262). Reel blinking pattern "3" involves continuous repetition of eleven types of illumination patterns within a travel time of 75.09 ms, to thereby cause blinking of the back lamps 125 (see Fig. 263). Reel blinking pattern "4" involves continuous repetition of nine types of illumination patterns within a travel time of 150.18 ms, to thereby cause blinking of

the back lamps 125 (see Fig. 264). Reel blinking pattern "5" involves continuous repetition of nine types of illumination patterns within a travel time of 103.25 ms, to thereby cause blinking of the back lamps 125 (see Fig. 265). Reel blinking pattern "6" involves continuous repetition of fifteen types of illumination patterns within a travel time of 150.18 ms, to thereby cause blinking of the back lamps 125 (see Fig. 266). Reel blinking pattern "7" involves continuous repetition of fifteen types of illumination patterns within a travel time of 75.09 ms, to thereby cause blinking of the back lamps 125 (see Fig. 267). Reel blinking pattern "8" involves continuous repetition of fifteen types of illumination patterns within a travel time of 103.25 ms, to thereby cause blinking of the back lamps 125 (see Fig. 268). In addition, there is a "No Blinking" pattern in which no blinking of the back lamps 125; that is, continuous illumination of the back lamps 125, is performed.

Winning determination data correspond to a flag for reporting the result of an internal generation operation performed by the control unit to the player with 100% confidence. Further, the "start sound" corresponds to a sound effect originating from the speaker 25 at the commencement of a game. In the present embodiment, two different types of start sounds; i.e., "Start Sound 1" and "Start Sound 2," are available. In the present embodiment, selection processing pertaining to the flash data tables is performed by the main control board 100. However, the sub-control board 200 may be configured to perform the processing.

<Image Display>

On the basis of the data constituting the foregoing tables, the image display section 13 displays images. Displayed images are expressed as stationary or animated. A character shows up and provides game rules or explanations of errors. Further, in association with a progress in an adventure story in which a main character appears, various presentations are effected.

Images appearing on the image display section 13 will be described by reference to Figs. 103 through 178.

Figs. 103 through 106 show demonstration display screens appearing in response to the previously-described demonstration display command code "02h" (see Fig. 28). These demonstration displays appear when no game is played on the slot machine. Sequentially displayed in an animated manner in accordance with an image sequence control table are images; for example, a manufacturer's logo (see Fig. 103), a model name (see Fig. 104), a dividend table (see Fig. 105), and explanations of game rules (see Fig. 106).

For instance, during the display of logo shown in Fig. 103, an archaeopteryx flies from the right side of the screen and enters a triangular frame displayed on the left-side of the screen, thereby completing a logo display. During the display of a model name shown in Fig. 104, model name "DUEL DRAGON" and a message indicating that insertion of gaming tokens is possible are displayed. Further, a dividend table of winning combinations which would arise during a normal game and a dividend table of winning combinations which would arise during bonus games are displayed in an animated manner. The dividend table shown in Fig. 105 states that, in a case where "DIAMOND," "DIAMOND," "DIAMOND" is displayed in the stopped state of the reels, fifteen gaming tokens are to be paid out. The game rule display shown in Fig. 106 explains that insertion of gaming token is necessary for starting a game. Subsequently, animated display of a round of games is scrolled.

Figs. 107 and 108 show reel screen displays appearing in response to the reel screen display command code "03h" (see Fig. 29). This reel screen display appears after insertion of a gaming token. The screen display is made up of a normal reel screen (see Fig. 107) and an internally-notified-state reel screen (see Fig. 108).

In the normal reel screen shown in Fig. 107 (in a case where 01h is stored in the second byte of the transmission command), the background is colored blue, thereby indicating that the display is a normal reel screen. In the internally-notified-state reel screen shown in Fig. 108 (in a case where 02h is stored in the second byte of the transmission command), the background is colored red.

Moreover, symbols; for example, "7," "7," "7," are displayed stationarily, thereby indicating that a big bonus is internally generated.

Figs. 109 through 120 show sign presentation screens appearing on the normal reel screen in response to the reel screen display command code "04h" (see Fig. 31). These sign presentation screens appear when the reels 5a to 5c start spinning on a normal reel screen. In these screens, there are displayed a plurality of types of small-jackpot combination sign presentations (corresponding to the transmission commands having data 00h to 07h stored in their second bytes) and a plurality of types of *LI-ZHI* sign presentations (corresponding to the transmission commands having 08h to 0Eh stored in their second bytes).

The sign presentation screen shown in Fig. 109 corresponds to a screen in which "DB, Cherry Winning L Sign (01h)" presentation is to be effected by means of the data stored in the second byte of the transmission command. The sign presentation screen shown in Fig. 110 corresponds to a screen in which "DB, Cherry Winning L Sign (02h)" presentation is to be effected. The sign presentation screen shown in Fig. 111 corresponds to a screen in which "Dragon Winning L Sign (03h)" presentation is to be effected. The sign presentation screen shown in Fig. 112 corresponds to a screen in which "Dragon Winning H Sign (04h)" presentation is to be effected. The sign presentation screen shown in Fig. 113 corresponds to a screen in which "Diamond Winning H Sign (05h)" presentation is to be effected. The sign presentation screen shown in Fig. 114 corresponds to a screen in which "Diamond Winning H Sign (06h)" presentation is to be effected. The sign presentation screen shown in Fig. 115 corresponds to a screen in which "Replay Winning Sign (07h)" presentation is to be effected.

The sign presentation screen shown in Fig. 116 corresponds to a screen in which "*OSHIKURA LI-ZHI* Sign (09h)" is to be performed. The sign presentation screen shown in Fig. 117 corresponds to a screen in which "PowerBall *LI-ZHI* Sign (0Ah)" is to be performed. The sign presentation screen shown in Fig. 118

corresponds to a screen in which "Balancing-on-Rolling-Ball *LI-ZHI* Sign (0Bh)" is to be performed. The sign presentation screen shown in Fig. 119 corresponds to a screen in which "YAH-HOO *LI-ZHI* sign (0Ch)" is to be performed. The sign presentation screen shown in Fig. 120 corresponds to a screen in which "BB Determination (Rainbow 7) (0Dh)" is to be performed.

Although unillustrated, a screen similar to the screen shown in Fig. 120 (in which "7," "7," "7" are changed to "BAR," "BAR," "BAR") appears in the "RB determination (rainbow BAR) (0Eh)" presentation.

Figs. 121 through 141 show *LI-ZHI* presentation screens to be displayed on the normal screen so as to correspond to the stop command code "07h" (see Fig. 34). The *LI-ZHI* presentation screen is displayed when an operation for stopping a second reel during play of a normal game is performed, in which a plurality of types of *LI-ZHI* presentations are effected.

For example, the *LI-ZHI* presentation screens shown in Figs. 121 and 122 are displayed when "Normal Winning (02h)" presentation is to be effected by means of the data stored in the second byte of the transmission command. When the three symbols stationarily displayed on the screen become identical (e.g., all are cookies), a message stating normal winning appears.

The *LI-ZHI* presentation screens shown in Figs. 123 through 126 are displayed when "*OSHIKURA* Winning (04h)" presentation is to be effected (Fig. 123), wherein the main character and a monster, who oppose each other, push each other with an apple sandwiched therebetween. In the case of "losing," the monster has won by pushing the main character away, and the main character is crushed by an apple that has fallen from above (see Fig. 124). Then, a "losing" symbol is displayed. In contrast, in the case of "winning," the main character has won by pushing away the monster, and the monster is crushed by the apple that has fallen from above (see Fig. 125). The apple located between them splits open, and "7" emerges from the inside of the split apple. Accordingly, the symbols assume

combination "7," "7," "7" (see Fig. 126).

The *LI-ZHI* presentation screens shown in Figs. 127 and 128 are displayed when "Balancing-on-Rolling-Ball Right Winning 1 (06h)" presentation is to be effected (Fig. 127). When the main character has successfully balanced on a rolling ball (see Fig. 127) and shown a V sign, with displayed symbols assuming combination "X," "X," "X," a message indicating "Winning" is displayed. In the case of "losing," the main character makes an unsuccessful landing when jumping down from the ball.

The *LI-ZHI* presentation screens shown in Figs. 129 through 131 are displayed when "Balancing-on-Rolling-Ball Left Winning 1 (0Ah)" presentation is to be effected. Although the main character has successfully balanced on a rolling ball (see Fig. 129), he ended with an unsuccessful landing (see Fig. 130). Later, when the main character wakes up and shows a V sign, with displayed symbols assuming combination "X," "X," "X," a message indicating "Winning" is displayed. In the case of "losing," the main character does not wake up.

The *LI-ZHI* presentation screens shown in Figs. 132 through 134 are displayed when "Balancing-on-Rolling-Ball Center Winning 1 (0Eh)" presentation is to be effected. After the main character has jumped on the ball (see Fig. 132) and landed on the ball (see Fig. 133), the thus-landed main character shows a V sign, with displayed symbols assuming combination "X," "X," "X." Then, a message indicating "Winning" is displayed. In the case of "losing," the main character makes an unsuccessful landing when jumping down from the ball.

The *LI-ZHI* presentation screens shown in Figs. 135 through 138 are displayed when "PowerBall 1 Winning (12h)" presentation is to be effected. The main character throws three powerballs sequentially toward a reel which is in motion and is to finally stop (i.e., with a symbol to be displayed on the center reel) (see Figs. 135 through 137). The three powerballs hit the target, and sign "HIT" appears (see Fig. 138), thereby indicating "Winning." In the case of "losing," no powerballs hit the target. "PowerBall 2 Winning (14h)" presentation is identical with "PowerBall

1 Winning," except that balls are thrown faster than in PowerBall 1.

The *LI-ZHI* presentation screens shown in Figs. 139 through 141 are displayed when "PowerBall 3 Winning (16h)" presentation is to be effected. The main character throws at the target a powerball, in which the character straining so as to attain the power for throwing three balls (see Figs. 139 and 140). The three powerballs hit the target, and sign "HIT" appears (see Fig. 141), thereby indicating "Winning." In the case of "losing," no powerballs hit the target.

Figs. 142 through 153 show big bonus stage presentation screens appearing in response to the bonus stage display instruction command code "0Ch" (see Fig. 39). The big bonus stage presentation screens appear during a big bonus game, wherein a story involving adventure of the main character is started.

The big bonus stage presentation screens shown in Figs. 142 and 143 are displayed when "BB Stage 1 Start Display (command code 02h is stored in the second byte of the transmission command)" presentation is to be effected. In association with start of a big bonus stage 1, message "Let's Go!" appears (see Fig. 142). In association with start of a normal game during a big bonus before a first JAC game is started, there is displayed a scene in which the main character goes to a green field (see Fig. 143).

The big bonus stage presentation screen shown in Fig. 144 is displayed when "BB Stage 2 Start Display (command code 03h)" presentation is to be effected. In association with start of a big bonus stage 2, there is displayed a scene in which the main character enters a forest.

The big bonus stage presentation screen shown in Fig. 145 is displayed when "BB Stage 3 Start Display (command code 04h)" presentation is to be effected. In association with start of a big bonus stage 3, there is displayed a scene in which the main character enters a cave. Thus, in the big bonus game, image presentations differ from stage to stage, thereby constituting a single story.

The big bonus stage presentation screens shown in Figs. 146 and 147 are

displayed when "Bonus End Presentation Display 1 (command code 05h)" presentation is to be effected. The main character knocks down a dragon, which is a character who is the boss of the main character's enemies, whereupon a message "See You!" is displayed (see Fig. 146). Subsequently, a curtain descends (see Fig. 147), and a message indicating the end of the big bonus game appears.

The big bonus stage presentation screens shown in Figs. 148 and 149 are displayed when "Bonus End Presentation Display 2 (RB1 and 2 ended with punctures) (command code 06h)" presentation is to be effected. The main character has failed to defeat an enemy monster (see Fig. 148) and has run away (see Fig. 149), thereby having failed to attain the greatest possible number of wins allowed in the first or second JAC game. Accordingly, the display shows that the game is ended with punctures (here, "PUNCTURE" means that the player has not won all the normal games available in a big bonus game).

The big bonus stage presentation screens shown in Figs. 150 and 151 are displayed when "Bonus End Presentation Display 3 (RB3 ended with punctures) (command code 07h)" presentation is to be effected. The main character has failed to defeat the dragon, the boss of the enemies, (see Fig. 150) and the main character's spirit rises to heaven (see Fig. 151). The display shows that the player has failed to attain the greatest possible number of wins allowed in the third JAC game and the bonus game has ended with so-called punctures.

The big bonus stage presentation screens shown in Figs. 152 and 153 are displayed when "Big Bonus End Presentation Display 4 (a normal game ended with punctures) (command code 08h)" presentation is to be effected. After a message "LOSS ONE'S WAY ("you've lost your way)" stating that the main character cannot meet the dragon has been displayed (see Fig. 152), another message "GAME OVER" is displayed (see Fig. 153), thereby showing that the player has failed to enter a JAC game during a normal game provided in the big bonus and has ended with so-called punctures.

The regular bonus stage presentation screens shown in Figs. 154 through 157 are displayed in response to the reel spinning start command code "0Dh" (see Fig. 40) for use in a regular bonus game. The regular bonus stage presentation screens are displayed during the course of a JACKPOT game. Accordingly, a story in which the main character experiences adventures proceeds.

For instance, the regular bonus stage presentation screens shown in Figs. 154 through 157 show a stage number in the JAC game (1 through 3) and the number of games provided in the JAC game (1 to 12), in association with progress of the adventure story. A stage number is represented by a background screen, such as a green field, a forest, or a cavern. Further, letter "TURN-1," "TURN-2," or the like appearing in the upper part on the screen denotes the number of games.

Jackpot winning presentation screens shown in Figs. 158 through 160 are displayed in response to the jackpot winning presentation command code "0Eh" (see Fig. 41). The jackpot winning presentation screen appears during the course of a JAC game, and a story in which the main character experiences adventures proceeds.

The jackpot winning presentation screens shown in Figs. 158 through 160 show a stage number (1 through 3) in the JAC game and the number of jackpot wins (1 through 8), in association with a progress in the adventure story. A stage number is represented by means of a background screen, such as a green field, a forest, or a cave. Letters "HIT-1" appearing in the top on the screen shows the number of jackpot wins.

Figs. 161 through 163 show the big bonus presentation screens shown in response to the reel spinning start command code "0Fh" (see Fig. 42) for use in a big bonus game. The big bonus presentation screens show the number of remaining games during the big bonus round.

On the big bonus presentation screens shown in Figs. 161 through 163, the number of remaining games provided in the big bonus is shown, by means of

numerals "3," "2," and "1." When a so-called punctured state approaches, a message to that effect is also displayed.

Figs. 164 through 169 show small-jackpot winning presentation screens which appear during a normal game provided in a big bonus game in response to the small-jackpot winning presentation command "10h" (see Fig. 43) for use in a normal game provided in a big bonus game. The small-jackpot winning presentation screens appear when small-jackpot winning has arisen during a normal game provided in a big bonus game. A story in which the main character experiences adventures proceeds.

For instance, on the small-jackpot winning presentation screen shown in Fig. 164, the main character picks up an object, and question mark "?" is provided in a thought balloon of the main character. This means that the current state is a so-called dropped state in which the player has failed to win in small-jackpot games even though a winning mode has been internally generated. The small-jackpot winning presentation screen shown in Fig. 165 shows success in so-called replay drop operation, by means of the word "NICE." Here, replay drop means that replay winning (indicated by JAC game operation symbols during a big bonus game) is dropped intentionally for the purpose of playing a specified number of normal games during the big bonus round.

On the small-jackpot winning presentation screens shown in Figs. 166 through 168, display of symbols "PB (ball)," "PB (ball)," and "PB (ball)" indicates occurrence of replay winning. Namely, there has arisen a specific winning mode for entering a JAC game. A character confronting the main character presents a stage number in the JAC game.

Symbols "DIAMOND," "DIAMOND," "DIAMOND" appear on the small-jackpot winning presentation screen shown in Fig. 169. Further, numeral "15" is displayed to indicate the number of gaming tokens to be paid out, thus displaying a winning mode of diamond.

Figs. 170 through 173 show regular bonus winning presentation screens which appear during a big bonus game in response to the regular bonus winning presentation command code "11h" (see Fig. 44) for use in a big bonus game. The regular bonus winning presentation screens are displayed when the player has won a regular bonus during the big bonus game, and a story of the main character experiencing adventures proceeds.

For instance, on the regular bonus winning (also called JAC game winning in some instances) presentation screens shown in Figs. 170 through 173, the main character confronts a monster (or dragon) and throws balls toward the monster. If a ball hits the monster, the monster changes into a token, and it is indicated that the player has won the jackpot game (bonus game) provided in the regular bonus game. The enemy is changed in accordance with a stage number in the regular bonus game. On the third stage, even if a ball hits the dragon, the dragon will not change into a token.

Figs. 174 through 178 show error display screens which are displayed in response to the error screen display command "12h" (see Fig. 45). The error display screens appear in the event of occurrence of an error in the slot machine. Details of the error and an error code are displayed along with characters.

For instance, the error display screen shown in Fig. 174 is displayed in the event of occurrence of a "selector jam error (01h is stored in the second byte of the transmission data)," in which a selector is jammed with gaming tokens. The error display screen shown in Fig. 175 is displayed in the event of occurrence of "token hopper full error (03h)," in which the token tray is filled with tokens. The error display screen shown in Fig. 176 is displayed in the event of occurrence of "Token-to-be-Replenished Error (05h)," in which shortage of gaming tokens has arisen in the hopper. The error display screen shown in Fig. 177 is displayed in the event of occurrence of "Hopper Jam Error (06h)," in which the hopper is jammed with gaming tokens. The error display screen shown in Fig. 178 is displayed in the

event of occurrence of "Board Anomalies Detected Error (07h)," in which anomalies have arisen in a board.

<Control Procedures>

As has been described above, the slot machine 1 is electrically controlled by means of control units (i.e., the main control board 100, the sub-control board 200, and the image control board 300).

Figs. 179 through 256 shows control operations of the slot machine 1, control operations to be performed by the sub-control board 200 and those to be performed by the image control board 300 are described with emphasis.

<Receive Interrupt Processing: A01>

Figs. 179 through 182 are flowcharts showing procedures for receive interrupt processing.

As shown in Figs. 179 through 182, receive interrupt processing is for enabling the sub-control board 200 to receive a signal from the main control board 100.

In receive interrupt processing, of received data, the higher eight bits are imported from the command receive port "COMHMAP (of higher hierarchical level)." The remaining, lower eight bits are imported from "COMLMAP (of lower hierarchical level)" (A01_1). Further, a strobe signal is imported from the input port [0] (INMAP0) (A01_2), thereby enabling multiple interrupts (timer interrupts) (A01_3).

Subsequently, the status of the strobe signal is fetched, and a determination is made as to whether or not the thus-imported signal is an invalid strobe signal, such as noise (A01_4). If the imported signal is an invalid strobe signal, it is determined that an error has arisen. Hence, the error counter "REER_CNT" is updated (A01_32), and the main CPU down detection time initial value (50 ms) shown in Fig. 47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area.

In contrast, if the imported signal is an effective strobe signal, the receive sequence management data "RX_PHASE" (a parameter for determining whether the strobe signal is a leading command or a subsequent command of a command which is transmitted in two bytes) (A01_5) are checked, thereby determining whether the signal corresponds to the first sequence or the second sequence (A01_6). The reason for this is that two bytes (one byte = eight bits) are taken as one sequence and that data are transmitted in the form of two sequences.

If the received sequence corresponds to the first sequence, the first byte (representing the type of a command code) of the received data is compared with the maximum value of the command code "MCMD_MAX" output from the main CPU 101 (A01_7), thereby determining whether or not the received data fall within the range of "1" to "MCMD_MAX (00FH)" (A01_8). Since the transmitted command code consists of sixteen types in total, the determination is made in order to prevent receipt of any further values (see Fig. 10). If the received data do not fall within the range of "1" to "MCMD_MAX (00FH)," it is determined that an error has arisen. The error counter "REER_CNT" is updated (A01_32). Further, the main CPU down detection time initial value (50 ms) shown in Fig. 47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area, thereby resulting in restoration from the receive interrupt processing.

If the received data do not fall within the range of "1" to "MCMD_MAX (00FH)," "1" indicating the end of the first sequence is set in the receive sequence management data "RX_PHASE" (A01_9). The received commands are saved in the received-command higher byte area "RCVCMDH" and in the received-command lower byte area "RCVCMDL" (A01_10).

Subsequently, BCC check is performed, and a computed BCC value is saved (A01_11). The timeout value "RX_TOUT" included in the received sequence shown in Fig. 47 is loaded, and the thus-loaded value is set in the receive timeout measurement timer "RX_TIMER" (A01_12). The main CPU down detection time

initial value (50 ms) shown in Fig. 47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area, thereby resulting in restoration from the receive interrupt processing.

If the received sequence is the second sequence, the receive sequence management data "RX_PHASE" is cleared (A01_13), and the current time of the receive timeout measurement timer "RX_TIMER" is checked. Then, the time value is compared with a time value of a receive interval timer RX_NTIM (A01_14). The receive timeout measurement timer "RX_TIMER" is cleared (A01_15), to thereby determine whether or not a receive interval is normal (A01_16).

If the receiver interval is normal BCC value of the receive command is checked (A01_17), to thereby determine whether or not a match exists between the computed BCC value and the received BCC value (A01_18).

If the match does not exist between the computed BCC value and the received BCC value, it is determined that error has arisen, thereby the error counter "REER_CNT" is updated (A01_32), and the main CPU down detection time initial value (50ms) shown in Fig.47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area, thereby resulting is restoration from the receive interrupt processing.

If the receiver interval is anomalous, it is determined that an error has arisen. The error counter "REER_CNT" is updated (A01_32). Further, the main CPU down detection time initial value (50 ms) shown in Fig. 47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area, thereby resulting in restoration from the receive interrupt processing.

If a match exists between the computed BCC value and the received BCC value (A01_18), the received command is deemed normal, thereby clearing the receive error counter "REER_CNT" (A01_19). On the basis of the information about the game status stored in the third byte of the received command (i.e., a higher byte stored in the second sequence), the game status flag "GAMEST" is

updated (A01_20). Further, "received-command higher byte area "RCVCMDH" (in which command code type information is stored) is checked (A01_21).

Subsequently, a determination is made as to whether or not the received command corresponds to the presentation initialization request (01H) (A01_22). If the received command corresponds to the presentation initialization request (01H), a presentation status flag "PRDC_STS" is checked (A01_23). Further, a determination is made as to whether or not an initialization command reject status is brought into an ON state and initialization is rejected (for example, as in the case where the initialization request has already been received) (A01_24). If the initialization request is accepted, a presentation initialization instruction flag of the presentation status flag "PRDC_STS" is set to ON. Further, an initialization command reject status flag is set to ON, thereby rejecting acceptance of another initialization command (A01_25). In a case where the initialization command reject status flag has already been set to ON (YES is selected in A01_24), the main CPU down detection time initial value (50 ms) shown in Fig. 47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area, thereby resulting in restoration from the receive interrupt processing.

If the received command does not correspond to the presentation initialization request (01H), the presentation initialization instruction flag of the presentation status flag "PRDC_STS" is cleared (A01_26). The command received this time is compared with a finally-received command area "LST_RCMD," thereby determining whether or not the commands are identical (A01_28). The main control board 100 transmits identical data several times in consideration of occurrence of failures to receive the data. If the thus-transmitted commands are received and saved, a plurality of identical commands are continuously saved. Eventually, a single presentation is repeated. In order to prevent such a situation, the received command is compared with the finally-received command area. More specifically, the command transmitted this time is transmitted repeatedly at an

interval of 50 ms or less until the next command is generated.

If the command received this time is identical with the finally-received command area "LST_RCMD," the main CPU down detection time initial value (50 ms) shown in Fig. 47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area, thereby resulting in restoration from the receive interrupt processing.

In contrast, if the command received this time is different from the finally-received command area "LST_RCMD" (i.e., the command received this time is a new one), a determination is made as to whether or not the received command is a sound presentation instruction command (A01_29).

If the command received this time differs from the sound presentation instruction command, the received command is saved as a finally-received command into "LST_RCMD" (A01_30). In contrast, if the command received this time is the sound presentation instruction command, step (A01_30) is skipped. For example, if tokens are inserted continuously, the main control board 100 transmits token insertion commands. In a case where an arrangement is made so as to monitor a final receipt command, if tokens are inserted continuously, the sub-control board 200 receives a token insertion command before the sequence of the token insertion command transmitted last time is completed. Accordingly, the token insertion command transmitted this time is ignored.

Received-command storage processing (i.e., processing F03 to be described in detail later) is performed (A01_31). The main CPU down detection time initial value (50 ms) shown in Fig. 47 is set in the main CPU down watchdog timer "M_WATCH" provided in the work area, thereby resulting in restoration from the receive interrupt processing.

As mentioned above, in contrast with a plurality of transmissions of a single command, if subsequently-transmitted commands should not be ignored (e.g., the sound presentation instruction command in the present embodiment), there is

performed processing for preventing disregard for subsequent commands, thus preventing interruption of a presentation.

<Timer Interrupts: A02>

Figs. 183 through 185 are flowcharts showing procedures for timer interrupt processing.

As shown in Figs. 183 through 185, timer interrupt processing involves execution of processing operations: that is, processing by which the sub-control board 200 transmits a signal to the image control board 300 after having received a signal from the main control board 100; updating of random numbers for selecting a presentation; monitoring of various errors; and updating of timers.

In the timer interrupt processing, multiple interrupts are enabled (A02_1), and the random numbers "SELRAND" are updated (A02_2). A timer for adjusting a presentation sequence "PR_TIMER" is checked (A02_3), thereby making a determination as to whether or not time-up has arisen in the timer "PR_TIMER" (A02_4). If time-up has not arisen in the timer "PR_TIMER," "one" is subtracted from a timer value of the presentation sequence adjustment timer "PR_TIMER" (A02_5). In contrast, if time-up has arisen in the timer "PR_TIMER," the timer value subtraction processing (A02_5) is not performed.

Subsequently, the receive timeout measurement timer "RX_TIMER" (for checking a time interval from receipt of the first sequence until receipt of the second sequence) is checked (A02_6), to thereby determine whether or not time-up has arisen in the receive timeout measurement timer "RX_TIMER" (A02_7). If time-up has not arisen in the timer "PX_TIMER," "one" is subtracted from a timer value of the "PX_TIMER" (A02_8). In contrast, if time-up has arisen in the timer "PX_TIMER," it is determined that an error has arisen. Received-sequence management data "RX_PAUSE" are cleared (A02_9) without performance of the timer value subtraction processing (A02_8). Further, the receive error counter "REER_CNT" is updated (A02_10).

A determination is made as to whether or not the main CPU down watchdog timer "M_WATCH" is checked (A02_11), to thereby determine whether or not time-up has arisen in the timer "M_WATCH" (A02_12). If time-up has arisen in the timer "M_WATCH," it is determined that the main CPU is down. The presentation initialization instruction flag of the presentation status flag "PRDC_STS" is set to ON (A02_14), thereby resulting in restoration from the timer interrupt processing.

If time-up has not arisen in the timer "M_WATCH," "one" is subtracted from a timer value of the "M_WATCH" (A02_13). A check is made as to transmission sequence management data "TX_PHASE" (a work area for managing whether or not a command is transmitted to the image control board 300) (A02_15), thereby determining whether or not data are being transmitted to the image control board 300 (A02_16). If data are being transmitted to the image control board 300, command transmission processing (i.e., processing A04 to be described in detail later) is performed (A02_17), thereby resulting in restoration from the timer interrupt processing.

If data are not transmitted to the image control board 300, a transmission management timer area "TX_TIMER" is checked (A02_18). A determination is made as to whether or not time-up has arisen in the transmission management timer area "TX_TIMER" (A02_19). This timer area is a work area for managing a transmission interval between transmission commands. In the present embodiment, the transmission management timer area is set to an initial value of 10 ms (see Fig. 47). Here, if time-up has arisen in the transmission management timer area "TX_TIMER," transmission start processing (i.e., processing A04 to be described in detail later) is performed (A02_20), thereby resulting in restoration from the timer interrupt processing.

In contrast, if time-up has not arisen in the transmission management timer area "TX_TIMER," "one" is subtracted from a timer value of the "TX_TIMER"

(A02_21), thereby checking the presentation status flag "PRDC_STS" (A02_22). A determination is made as to whether or not the transmission command has already been edited (A02_23). This flag means that various image presentations are selected in accordance with the presentation command output from the main control board 100, whereby the command to be transmitted to the image control board 300 is determined (i.e., edited).

If the transmission command has not yet been edited, restoration from the timer interrupt processing is effected.

In contrast, if the transmission command has already been edited, the transmission data saved in a transmission command edition buffer (doubling as a backup buffer) "TXBUFWK" are set in a transmission command buffer "TXBUFF" (A02_24). The game status flag "GAMEST" is checked (A02_25), thereby determining whether or not an error has arisen (A02_26).

In the event that an error has arisen, the transmission data saved in a dedicated buffer for an error screen display command (doubling as a backup buffer) "TXERRWK" are set in the transmission command buffer "TXBUFF" (A02_27). In contrast, if no error has arisen, step (A02_27) is skipped.

Subsequently, computation of a BCC value of the transmission command is set (A02_28), and a transmission command edited flag of the presentation flag "PRDC_STS" is cleared (A02_29), thus resulting in restoration from the timer interrupt processing.

<Transmission Start Processing: A03>

Fig. 186 is a flowchart showing procedures for transmission start processing.

As shown in Fig. 186, transmission start processing is for enabling the sub-control board 200 to start transmission of a signal to the image control board 300.

In transmission start processing, the transmission command buffer "TXBUFF" is first checked (A03_1), thereby determining whether or not a

transmission command is stored (A03_2). If no transmission command is saved, restoration from the transmission start processing is effected.

If a transmission command is stored in the buffer, a transmission counter "TXDATCNT" is updated by the amount corresponding to the number of transmission data sets (A03_3). "1" is set in the transmission sequence management data "TX_PHASE," thereby indicating that transmission is in progress (A03_4). A transmission start code "DSP_STX" is transmitted (A03_5), and restoration from transmission start processing is effected.

<Command Transmission Processing: 04>

Fig. 187 is a flowchart showing procedures for command transmission processing.

As shown in Fig. 187, command transmission processing is for enabling the sub-control board 200 to send a signal to the image control board 300.

In command transmission processing, the data saved in the transmission command buffer "TXBUFF" are output to the OUT port and transmitted (A04_1). The transmission counter "TXDATCNT" is decremented by the amount corresponding to the number of transmission data sets (A04_2). A determination is made as to whether or not the transmission counter "TXDATCNT" has performed a count-up operation (A04_3).

If the transmission counter "TXDATCNT" has performed a count-up operation, "0" is set in the transmission sequence management data "TX_PHASE," thereby indicating that transmission is stopped (A04_4). A command transmission interval "TX_INTVAL" is set in the transmission management timer "TX_TIMER" (A04_5), thereby resulting in restoration from the command transmission processing.

In contrast, if the transmission counter "TXDATCNT" has not performed a count-up operation, transmission of the data stored in the transmission command buffer "TXBUFF" (A04_1 through A04_3) is continued.

<Presentation Control Processing (main loop): C01>

Figs. 191 and 192 are flowcharts showing procedures for presentation control processing.

As shown in Figs. 191 and 192, presentation control processing is a main loop for enabling the image display section 13 to perform presentation control operation.

In the presentation control processing, the watchdog timer "M_WATCH" is reset (C01_1), and the receive error counter "REER_CNT" is checked. A count value of the counter is compared with the permissible number of consecutive receiver errors "ERRN_MAX" (C01_2), thereby determining whether or not the number of errors has exceeded an upper limit value 0F (16) (C01_3).

If the count value of the receive error counter "REER_CNT" exceeds 16; that is, if errors have arisen continuously and restoration from an error state has not been effected, acceptance of interrupts is inhibited (C01_16). Further, the presentation sequence timing adjustment timer "PR_TIMER" is cleared (C01_17), and the receive error counter "REER_CNT" is also cleared (C01_18). The receive sequence management data "TX_PHASE" are cleared, thereby forcefully terminating transmission (C01_19). Further, the receive sequence management data "RX_PHASE" are cleared, thereby forcefully terminating a receiving operation (C01_20). The finally-received command area "LST_RCMD" is cleared (C01_21), and a backup data check code "BKCK_DAT" is brought into an ON state, thereby indicating backup anomalies (C01_22). Processing proceeds to presentation restoration check processing (processing B01 to be described later).

In contrast, if the count value of the receive error counter "REER_CNT" does not exceed 16; that is, if errors have not arisen continuously and restoration from an error state has been effected, the presentation status flag "PRDC_STS" is checked (C01_4), thereby determining whether or not the presentation status flag is an initialization instruction (C01_5).

If the presentation status flag is an initialization instruction, processing analogous to that performed when the count value of the receive error counter "REER_CNT" has exceeded 16 (C01_16 through C01_22) is performed, whereby processing proceeds to presentation restoration check processing (i.e., processing B01 to be described later).

If the presentation status flag is not an initialization instruction, the presentation status flag "PRDC_STS" is checked (C01_6), determining whether or not the transmission command has already been edited; i.e., whether or not there is a command to be transmitted to the image control board 300 (C01_7). If the transmission command has not yet been edited, a check is made to the number of received data sets "RECCNT" (C01_8), thereby determining whether or not a presentation instruction is output from the main CPU 101 (C01_09).

If a presentation instruction is output from the main CPU 101, there are performed checking of a gaming status (i.e., processing C02 to be described in detail later) (C01_10), fetching of a command (i.e., processing F04 to be described in detail later) (C01_11), and analysis of a received command (i.e., processing D01 to be described in detail later) (C01_12).

Subsequently, if the transmission command has already been edited (YES is selected in C01_7) and no presentation instruction is output from the main CPU 101 (NO is selected in C01_9), the presentation status flag "PRDC_STS" is checked (C01_13), thereby determining whether or not sequence control operation is in progress (C01_14). Here, a sequence control operation which is in progress specifically means that sound effects are being generated. For instance, the flag shows that the *LI-ZHI* presentation sequence control table shown in Fig. 53 is in progress (i.e., a *LI-ZHI* presentation is being performed).

If the sequence control is in progress, presentation sequence control processing (processing C03 to be described in detail later) is performed (C01_15), and processing proceeds to an initial step of the presentation control processing.

<Presentation Restoration Check Processing: B01>

Figs. 188 and 189 are flowcharts showing procedures for presentation restoration check processing.

As shown in Figs. 188 and 189, the presentation restoration processing is performed when it is found that the command transmitted from the main control board 100 is an initialization request or when a game is resumed after having been interrupted for reasons of an error such as an empty hopper.

In the presentation restoration check processing, presentation status reset processing (i.e., F01 processing to be described in detail later) is performed, thereby initializing a presentation status (B01_1). The presentation initialization instruction flag of the presentation flag "PRDC_STS" is cleared, and the initialization command reject status flag is set to ON, thereby rejecting initialization (B01_2). A command transmission internal "TX_INTVAL (10 ms)" at which a command is to be transmitted to the image control CPU is set in the transmission management timer "TX_TIMER" (B01_3). Further, the receive error counter "REER_CNT" is set in the register (B01_4), thereby enabling interrupts (B01_5).

Subsequently, the watchdog timer "M_WATCH" is reset (B01_6), and the backup data check code "BKCK_DAT" is checked (B01_7), thereby determining whether or not a backup operation is operated normally (B01_8). If a backup operation is not performed normally, processing proceeds to presentation nonrestoration processing (i.e., processing B02 to be described later).

In contrast, if a backup operation is performed normally, sound restoration processing (i.e., processing F02 to be described in detail later) is effected (B01_9). The presentation status flag "PRDC_STS" is checked (B01_10), thereby determining whether or not command analysis is in progress (B01_11).

In a case where command analysis is in progress, commands "ALCMD_HI" and "ALCMD_LO," which are being analyzed, are set in the register (B01_12), where the commands are subjected to received-command analysis processing (i.e.,

processing D01 to be described later) (B01_13). Processing then proceeds to presentation control processing (i.e., processing C01 to be described in detail later).

In contrast, if command analysis is not in progress, the transmission command edited flag of the presentation status flag "PRDC_STS" is set to ON (B01_14). The presentation status flag "PRDC_STS" is checked (B01_15), thereby determining whether or not sequence control is in progress (B01_16).

In a case where sequence control is in progress, a presentation-sequence-control-table-access pointer backup "PRSQTBK" is set in the presentation-sequence-control-table-access pointer "PRSQPTR." The remaining portion of the sequence control which has not yet been processed as a result of occurrence of an error is resumed (B01_17), and processing proceeds to presentation control processing (i.e., processing C01 to be described in detail later). In a case where sequence control is not in progress, processing jumps directly to presentation control processing (i.e., processing C01 to be described in detail later).

<Presentation Nonrestoration Processing: B02>

Fig. 190 is a flowchart showing procedures for presentation nonrestoration processing.

Presentation nonrestoration processing is effected when processing cannot return to a presentation status before occurrence of an error, for reasons of corruption of backup data. As shown in Fig. 190, this processing is for shifting processing to received-command analysis processing in a case where backup has not been performed normally during the presentation restoration processing.

In presentation nonrestoration processing, the watchdog timer "M_WATCH" is reset (B02_1), and a RAM location is cleared (B02_2). Interrupts are enabled (B02_3), the game status flag "GAMEST" is checked (B02_4), and a determination is made as to whether or not internal winning is being notified (B02_5).

If internal winning is being notified, the number of games "PCHB_NUM" for changing symbols to be stationarily displayed after illumination of the WIN lamps

shown in Fig. 46 is set in a displayed symbol change counter "WPLY_CNT"(for notifying the number of games after which the type of bonus game is to be notified when a big bonus game or a regular bonus game has not yet been determined even though internal winning to be described later is being notified), and the counter is initialized (B02_6). The reel screen type flag of the presentation status flag "PRDC_STS" is set to ON, thereby indicating that an internally-notified state has arisen (B02_7). There are selected an initial value for data 1 pertaining to a symbol to be displayed on a liquid-crystal screen (simply called "liquid-crystal screen displayed-symbol data 1") for effecting the internal winning shown in Fig. 46, and an initial value for data 2 pertaining to a symbol to be displayed on a liquid-crystal screen (simply called "liquid-crystal screen displayed-symbol data 2") for effecting the same (B02_8). If internal winning is not notified, an initial value for the liquid-crystal screen displayed-symbol data 1 and an initial value for the liquid-crystal screen displayed-symbol data 2 are selected (B02_9). In other words, in the presentation restoration processing (i.e., when backup is performed normally), displayed symbol data are stored in a center-side displayed-symbol save area "SREEL_BK" and a right-side displayed-symbol save area "RREEL_BK." However, the presentation nonrestoration processing involves selection of initial values for the displayed-symbol data.

Selected initial values are set to the stationarily-displayed symbol data 1 "STP_PIC1," stationarily-displayed symbol data 2 "STP_PIC2," displayed-symbol data 1 "DSP_PIC1," and displayed-symbol data 2 "DSP_PIC2" (B02_10). The commands "ALCMD_HI" and "ALCMD_LO," which are being analyzed, are set in the register (B02_11). A received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is set to ON (B02_12), thereby effecting received-command analysis processing (i.e., processing D01 to be described in detail later).

<Game Status Check Processing: C02>

Figs. 193 through 195 are flowcharts showing procedures for game status check processing.

As shown in Figs. 193 through 195, the game status check processing is for generating a presentation status (e.g., sound effects and an image to be displayed on the liquid-crystal screen) corresponding to the status of a game by means of checking the current state of a game.

In the game status check processing, the game status flag "GAMEST" is checked (C02_1), and a sound backup area 1 "SND_BAK" is checked (C02_2), thereby determining whether or not a play-out sound is being generated (C02_3). A sound backup area consists of four bytes in total and backs up sound data corresponding to playback channels (CH1 through CH4) (please refer to Figs. 61 through 75 for correspondence between sound effects and playback channels).

If a play-out sound is being generated, a determination is made as to whether or not the slot machine is in a play-out state (C02_4). If the slot machine is not in the play-out state, it is considered that the slot machine is in a state other than the play-out state or is released from the play-out state. Hence, a play-out sound mute code "SD_OFF1" [since a play-out sound is played back by way of the channel 1 (see Fig. 63), the code is a channel 1 mute code] is set in the register (C02_6), and the sound backup area 1 "SND_BAK" is cleared (C02_7).

If the play-out sound is not being generated, the sound backup area 1 "SND_BAK" is checked (C02_8), thereby determining whether or not a bonus sound is being generated (C02_9). In a case where a bonus sound is being generated, the game status flag "GAMEST" is further checked (C02_10), thereby determining whether or not a bonus game is being played (C02_11).

If a bonus game is not being played, a bonus sound mute code "SD_OFF1" is set in the register (C02_12), and sound control processing (i.e., processing G01 to be described in detail later) is performed (C02_13). Thus, the sound backup area 1 "SND_BAK" is cleared (C02_14).

If a bonus game is being played (YES is selected in C02_11), if the slot machine is not in a play-out state (C02_5 to C02_7), or if no bonus sound is output (NO is selected in C02_9), the sound backup area 1 "SND_BAK" is checked (C02_15), thereby determining whether or not an error sound is being output (C02_16).

If an error sound is output, the game status flag "GAMEST" is checked (C02_17), thereby determining whether or not an error sound is being generated (C02_18). If no error sound is generated, the play-out sound mute code "SD_OFF1" is set in the register (C02_19), and sound control processing (i.e., processing G01 to be described in detail later) is performed (C02_20). The sound backup area 1 "SND_BAK" is cleared (C02_21).

Subsequently, if no error sound is generated (NO is selected in C02_16) or if an error sound is generated (YES is selected in C02_18), the game status flag "GAMEST" is checked (C02_22), thereby determining whether or not internal winning is being notified (C02_23).

In a case where internal generation of a winning mode is being notified, processing returns to the main routine. In contrast, if internal winning is not notified, the presentation status flag "PRDC_STS" is checked (C02_24), thus determining whether or not a normal reel screen is displayed (C02_25). If a normal reel screen is displayed, processing returns to the main routine.

In contrast, if a normal reel screen is not displayed, internal winning is not notified on the liquid-crystal screen in spite of an internal winning notification state (i.e., WIN lamps are illuminated), thus resulting in generation of presentation mismatch (when the WIN lamps are illuminated; that is, when the main control board 100 has selected determination data, an internally-generated-bonus determination screen inevitably appears on the liquid-crystal screen). Hence, the image is initialized. More specifically, the reel screen type flag of the presentation status flag "PRDC_STS" is set to OFF (C02_26). The displayed-symbol data 1 initial

value "PIC_INIT1" is set in the displayed-symbol data "DSP_PIC1" (C02_27). The displayed-symbol data 2 initial value "PIC_INIT2" is set in the displayed-symbol data "DSP_PIC2" (C02_28). Then, processing returns to the main routine.

<Presentation Sequence Control Processing: C03>

Figs. 196 and 197 are flowcharts showing procedures for presentation sequence control processing.

As shown in Figs. 196 and 197, presentation sequence control processing is for effecting presentation sequence by means of making a determination as to a timing at which the presentation sequence is to be effected. In the present embodiment, the presentation sequence primarily means sound-effect (sound) sequence.

In presentation sequence control processing, the game status flag "GAMEST" is checked (C03_1), thereby determining whether or not errors continue to occur (C03_2). Here, if errors continue to occur, normal sequence control is impossible, and hence processing returns to the main routine.

In contrast, if there are no errors, the presentation sequence timing adjustment timer "PR_TIMER" (C03_3) is checked, thereby determining whether or not time-up has arisen in the presentation sequence timing adjustment timer "PR_TIMER"; that is, whether or not a sequence implementation timing has arisen (C03_4). If no sequence implementation timing has arisen, processing returns to the main routine.

If the sequence implementation timing has arisen, the presentation-sequence-access pointer "PRSQPTR" is checked "C03_5," thereby determining whether or not data stored in a corresponding address of the sequence control table are a sequence end code "ENDSQ" (C03_6). If the data are a sequence end code "ENDSQ," one presentation sequence control operation has already been completed. Hence, a *LI-ZHI*-presentation-in-effect flag of the presentation status flag "PRDC_STS" is cleared, and the

presentation-sequence-control-in-progress flag is cleared, thus indicating end of the sequence control operation (C03_7). The game status flag "GAMEST" is checked (C03_8), thereby determining whether or not the slot machine is in a play-out state (C03_9). If the slot machine is not in a play-out state, processing returns to the main routine.

If the slot machine is in a play-out state, the play-out sound code "SD_OVER" is set in the register and in the sound backup area 1 "SND_BAK" (C03_10), and sound control processing (i.e., processing G01 to be described in detail later) is effected (C03_11).

If the data stored in the address represented by the presentation-sequence-access pointer "PRSQPTR" are not the sequence end code "ENDSQ" (NO is selected in C03_6), a determination is made as to whether or not the address data indicated by the presentation-sequence-access pointer "PRSQPTR" are a sequence repeat code "REPSQ" (C03_12).

If the address data indicated by the presentation-sequence-access pointer "PRSQPTR" are a sequence repeat code "REPSQ," the data pertaining to the sequence control table (i.e., a leading address of the presentation sequence control table) indicated by the address stored in the sequence-control-table-access pointer backup "SQPTRBK" are set in the register as a sound control request code (C03_13). In contrast, if the address data indicated by the presentation-sequence-access pointer "PRSQPTR" are not a sequence repeat code "REPSQ," step (C03_13) is skipped.

Subsequently, from the data stored in the currently-selected address, a determination is made as to whether or not sound is being generated (C03_14). If sound is being generated, the sound data are set in the register (C03_15), and sound control processing (i.e., processing G01 to be described in detail later) is effected (C03_16). In contrast, if no sound is generated, processing (C03_16) is skipped.

Subsequently, the address of the sequence-control-table-access pointer

"PRSQPTR" is updated to the next address (C03_17), and data pertaining to the thus-updated address are set in the presentation-sequence-timing adjustment timer "PR_TIMER" (C03_18). Processing is then returned.

<Received Command Analysis Processing: D01>

Fig. 198 is a flowchart showing procedures for received-command analysis processing.

As shown in Fig. 198, received-command analysis processing is for enabling analysis of a command stored in the register through use of command fetch processing (i.e., processing F04 to be described in detail later) and enabling a jump to corresponding processing.

In received-command analysis processing, the command data stored in the register are analyzed (D01_1). A jump is made to corresponding processing by reference to the result of analysis and a branch table (D01_2). The branch table is prepared from a demonstration display command code of the command received from the main CPU and shown in Fig. 10, through use of fourteen sub-routines provided so as to correspond to a sound presentation instruction command code. A received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Demonstration Display Command Processing: D02>

Fig. 199 is a flowchart showing procedures for demonstration display command processing.

As shown in Fig. 199, demonstration display command processing is for enabling the image display section 13 to control a demonstration display.

In demonstration display command processing, a demonstration display command "DSP_DEMO" is set in the transmission-command-edition buffer "TXBUFWK" (D02_1), thereby clearing a demonstration-display-in-progress flag of the presentation status flag "PRDC_STS," and the transmission-command-edited flag is set to ON (D02_2). The received-command-being-analyzed flag of the

presentation status "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Token Insertion Command Processing: D03>

Figs. 200 and 201 are flowcharts showing procedures for token insertion command processing.

As shown in Figs. 200 and 201, token insertion command processing is for controlling an image display to be displayed by the image display section 13 and generation of sound effects.

In token insertion command processing, a register value is set to a token counter "MEDLCTR" (D03_1), and the token insertion sound code shown in Fig. 58 "SD_MIN" is set in the register (D03_2). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D03_3), and the game status flag "GAMEST" is checked (D03_4). Thus, a determination is made as to whether or not a bonus game is being played (D03_5).

If a bonus game is being played, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine. The reason for this is that, during the course of the bonus game, a demonstration display is not effected and there is no necessity for changing a display on the liquid-crystal screen by means of insertion of a gaming token (a bet operation). A token insertion sound is controlled by means of sound single processing to be described later. In contrast, if no bonus game is being played, the presentation status flag "PRDC_STS" is checked (D03_6), thereby determining whether or not a *LI-ZHI* presentation is in operation (D03_7).

Even when a *LI-ZHI* presentation is in operation, a screen display is not changed by means of insertion of a gaming token, and the received-command-analysis-in-progress flag of the presentation status "PRDC_STS" is cleared (D01_3), and processing returns to the main routine. In contrast, if no *LI-ZHI* presentation is in operation, the presentation status flag "PRDC_STS" is

checked (D03_8), thereby determining whether or not a normal reel screen is being displayed (D03_9).

When a normal reel screen is being displayed, a display designation code "NORM_REEL" for a normal reel screen is selected (D03_10). When a normal reel screen is not being displayed, a display designation code "FGDU_REEL" for an internally-notified-state reel screen is selected (D03_11).

Subsequently, the gaming token insertion command code "03h" is set, as a first byte, in the transmission-command-edition buffer "TXBUFWK." Further, the selected reel screen type is set as a second byte, and the displayed-symbol data 1 "DSP_PIC1" are set as a third byte. The displayed-symbol data 2 "DSP_PIC2" are set as a fourth byte (D03_12).

Since the slot machine is in neither a bonus game nor a *LI-ZHI* presentation, a demonstration display may be displayed. During a demonstration display, there is a necessity for changing the screen display to a reel screen by means of inserting a gaming token. Hence, the demonstration display flag of the presentation status flag "PRDC_STS" is set to ON (C03_14), and the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Processing for Starting a Game during a Normal Game: D04>

Fig. 202 is a flowchart showing procedures pertaining to processing for starting a game during a normal game (simply called "game start command processing").

As shown in Fig. 202, game start command processing is for enabling the image display section 13 to start presentation display corresponding to a normal game or an internally-generated bonus game.

In the game start command processing, the reel stop counter "STOPCTR" is cleared (D04_1), and an internally-generated combination is stored in a generated-combination-type "WAVEBIT" on the basis of the data stored in the

register (D04_2). The game status flag "GAMEST" is checked (D04_3), thereby determining whether or not a bonus game is being played (D04_4).

If a bonus game is not being played, a presentation-in-effect flag "PRSELFLG" is set to ON (D04_5), the received-command-analysis-in-progress command of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine. In a case where a bonus game is being played, processing (D04_5) is skipped.

<Processing for Starting Spinning of Reels during a Regular Bonus: D05>

Figs. 203 and 204 are flowcharts showing procedures pertaining to processing for starting spinning of reels during a regular bonus game (called "RB spinning start command processing").

As shown in Figs. 203 and 204, the RB spinning start command processing is for enabling the image display section 13 to start presentation display corresponding to a regular bonus game.

In the RB spinning start command processing, the reel stop counter "STOPCTR" is cleared (D05_1), and the data stored in the register are analyzed (D05_2). The reel-spinning-start command code "0Dh" for a regular bonus game is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (D05_3). A regular bonus stage is set in the second byte of the transmission-command-edition buffer "TXBUFWK" (D05_4). A determination is made as to whether or not the regular bonus game originates from a normal game or a big bonus game (D05_5) (in the case of a single regular bonus game originating from a normal game, data pertaining to the number of possible regular bonus operations are set to "0").

If the regular bonus game originates from a big bonus game, regular-bonus-stage data are set in bonus-stage-data "BNS_STGN" (D05_6). In contrast, if the regular bonus game originates from a normal game, processing (D05_6) is skipped.

The number of available regular bonus games; that is, the number of jackpot games available during the regular bonus game (usually set to an initial value of 12), is set in an area concerning the number of available regular bonus games "JACGAME" (D05_7). A determination is made as to whether or not a stage number of the regular bonus is 3; that is, whether or not the regular bonus is on the final stage (D05_8).

If the regular bonus is not in the final stage, an operation sound code "SD_RBBGM1" for stage "1" or "2" of the regular bonus game is selected (D05_9). Further, a sequence-control-address table "RBP12_TBL" for stage "1" or "2" of the regular bonus game is selected (D05_10).

If the regular bonus is in the final stage, an operation sound code "SD_RBBGM3" for final stage "3" of the regular bonus game is selected (D05_11). Further, a sequence-control-address table "RBP3_TBL" for final stage "3" of the regular bonus game is selected (D05_12).

As mentioned above, various presentations are feasible by means of changing BGM from stage to stage during a regular bonus game. Further, the player can correctly ascertain a progress in the current bonus game.

An address of the selected sequence control table is set in the sequence-control-table-access pointer "PRSQPTR" and in the sequence-control-table-access pointer backup "SQPTRBK" (D05_13). The sequence-timing-adjustment timer "PR_TIMER" is cleared (D05_14). The transmission-command-edited flag and the sequence-control-in-progress flag of the presentation status flag "PRDC_STS" are set to ON (D05_15). A reel spinning start sound 1 code "ST_STT1" is set in the register (D05_16), and sound control processing (processing G01 to be described in detail later) is effected (D05_17).

The bonus operation sound backup area 1 "SND_BAK" is checked (D05_18), thereby determining whether or not the selected regular bonus operation sound is being generated (D05_19). In a case where the thus-selected regular bonus

operation sound is currently being generated, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

The foregoing processing is performed for the following reasons: Namely, this sub-routine is executed every time a game is started during a regular bonus game (i.e., every time the main control board 100 transmits the RG spinning start command). In a case where a regular bonus operation sound is currently being generated as a result of the previous game, if a regular bonus operation sound request is again issued to the sound-source IC 206 in the current game because the player is currently playing a regular bonus game, the currently-generated sound will be interrupted, and the same sound will be generated again from the beginning, thus breaking smooth, seamless presentation (the same also applies to processing for a game during a big bonus, which will be described later).

If the selected regular bonus sound is not being generated now, the selected regular bonus operation sound code is set in the bonus-operation-sound backup area 1 "SND_BAK" (D05_20). The selected regular bonus operation sound code is set in the register (D05_21). Sound control processing (i.e., processing G01 to be described later) is effected (D05_22). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Processing for Starting a game during a Big Bonus: D06>

Figs. 205 and 206 are flowcharts showing procedures pertaining to command processing for starting spinning of reels during a big bonus game (called "BB spinning start command processing").

As shown in Figs. 205 and 206, BB spinning start command processing is for enabling the image display section 13 to start presentation display corresponding to a big bonus game.

In the BB spinning start command processing, the data stored in the register

are analyzed (D06_1). The reel-spinning-start command code "0Fh" for a big bonus game is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (D06_2). A big bonus stage is computed in accordance with descriptions of the register (D06_3), and big-bonus-stage data are set in bonus stage data "BNS_STGN" (D06_4), thereby determining whether or not the stage of the big bonus game is in stage "3"; that is, whether or not the big bonus is in the final stage (D06_5).

If the big bonus game is in the final stage, a big-bonus-operation-sound code "SD_BBBGM3," which is to be effected in the final stage of the big bonus and is shown in Fig. 58, is selected (D06_6). In contrast, if the big bonus game is not in the final stage, a determination is made as to whether a symbol on the spinning reel is "White 7" or "Red 7" (D06_7). If the symbol is "White 7," a big-bonus-operation sound code "SD_BBBGM1" to be effected to "White 7" is selected (D06_8). If the symbol is "Red 7," a big-bonus-operation sound code "SD_BBBGM2" to be effected to "Red 7" is selected (D06_9).

Subsequently, the number of remaining big bonus games is set in the second byte of the transmission-command-edition buffer "TXBUFWK" (D06_10). The number of remaining big bonus games "BBPCTR" is updated (D06_11), and the transmission-command-edited flag and the sequence-control-in-progress flag of the presentation status flag "PRDC_STS" are set to ON (D06_12). A reel-spinning-start sound 1 code "SD_STT1" shown in Fig. 58 is set (D06_13), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D06_14).

Subsequently, the number of remaining big bonus numbers "BBPCTR" is checked (D06_15), thereby determining whether or not the number of remaining games is "five" or less (D06_16). If the number of remaining games is "five" or less, a countdown sound code matching the number of remaining games, such as that shown in Fig. 60, is set (D06_17), and sound control processing (i.e., processing G01

to be described in detail later) is effected (D06_18). In contrast, if the number of remaining games is "5" or more, the foregoing processing operations (D06_17 and D06_18) are skipped.

If the number of remaining normal games provided during the big bonus game decreases, a countdown presentation is performed in accordance with the state of the game, thereby enhancing the feeling of impatience or expectation (e.g., the player thinks "The game will be ended with punctures if I lose these three remaining games. I hope a combination of symbols for replay operation (i.e., a combination of symbols for effecting a JACKPOT game) shows up immediately !). Thus, the entertainment value of a game can be enhanced.

The sound backup 1 "SND_BAK" is checked (D06_19), thus determining whether or not the selected big bonus sound is currently being generated (D06_20). If the selected big bonus sound is currently being generated, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

In contrast, if the selected big bonus sound is not currently being generated; for example, in the case of a first game appearing after a stage of the big bonus game has changed to another stage, the selected big bonus operation sound code is set in the bonus-operation-sound backup area "SND_BAK" (D06_21). The selected big bonus operation sound code is set in the register (D06_22), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D06_23). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Stop Reel Command Processing: D07>

Figs. 207 through 210 are flowcharts showing procedures for stop reel command processing.

As shown in Figs. 207 through 210, stop reel command processing enables the image display section 13 to control presentation display and sound effects so as

to correspond to the reels 5a to 5c that have come to a stop.

In stop reel command processing, the data stored in the register are first analyzed (D07_1), thereby determining whether or not the first reel has stopped (D07_2). If the first reel has stopped, a first reel stop sound code "SD_STP1" shown in Fig. 59 is selected (D07_3). Next, a determination is made as to whether or not the second reel has stopped (D07_4). If the second reel has not yet stopped, a third reel stop sound code "SD_STP3" is selected (D07_5). If the second reel has stopped, a second reel stop sound code "SD_STP2" is selected (D07_6). If the first reel has not yet stopped (NO is selected in D07_2), processing (D07_3 through D07_6) is skipped.

Next, the reel stop counter "STOPCTR" is updated (in this case, the counter is brought to a standstill) (D07_7), and the selected stop sound code is set in the register (D07_8). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D07_9).

Subsequently, the game status flag "GAMEST" is checked (D07_10), thereby determining whether or not a bonus game is being played (D07_11). When the bonus game is being played, the received-command-analysis-in-progress flag of the presentation flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

When the bonus game is not being played, a sign presentation type "PRE_CLS" is checked (D07_12), thus determining whether or not the bonus determination sign (i.e., a balloon *LI-ZHI*) is being displayed (D07_13). If the bonus determination sign is being displayed, the received-command-analysis-in-progress flag of the presentation flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

In contrast, if the bonus determination sign is not being displayed, the reel stop command "07h" for a normal reel screen is selected as a transmission command (D07_14). The presentation status flag "PRDC_STS" is checked while the *LI-ZHI*

presentation type "RECH_CLS" is set in the register (D07_15 and D17_16).

In accordance with the result of checking, a determination is made as to whether or not the internally-notified-state reel screen is currently being displayed (D07_17). If the reel screen is being displayed, the stop command "08h" for an internally-notified-state reel screen is selected (D07_18). Since the internal generation of a winning mode has already been internally notified, there is no necessity for performing a *LI-ZHI* presentation. Hence, memory contents of the register are cleared (without a *LI-ZHI* presentation) (D07_19). In contrast, if the internally-notified-state reel screen is not being displayed, processing (D07_18 and D7_19) is skipped.

As mentioned above, an arrangement is made so as to prevent performance of a *LI-ZHI* presentation when the internally-notified-state reel screen is already displayed (i.e., when internal-generation of bonus winning has been determined and displayed). Thus, there can be prevented occurrence of an unfavorable sensation in the player, which would otherwise be caused by unnecessary presentation (for example, an unfavorable sensation of wondering why *LI-ZHI* presentation is performed after some time has elapsed since generation of bonus winning was determined).

On the basis of the result of processing, the selected reel screen type is set in the first byte of the transmission-command-edition buffer "TXBUFWK"; and memory contents of the register (i.e., a *LI-ZHI* presentation type) are set in the second byte (D07_20). The stationarily-displayed symbol data 1 "STP_PIC1," the stationarily-displayed symbol data 2 "STP_PIC2," and the sequence number for which the current stopping operation is to be performed (i.e., the first stoppage, the second stoppage, or the third stoppage) are set in the third and fourth bytes (D07_21 and D07_22). The reel stop counter "STOPCTR" is checked (D07_23).

In accordance with the result of checking, a determination is made as to whether or not the current stopping operation is the first stoppage (D07_24). If the

stopping operation is the first stoppage, it is too early to perform *LI-ZHI* presentation even if a *LI-ZHI* presentation is to be performed. Hence, the OFF state of the *LI-ZHI* presentation status flag of the presentation status flag "PRDC_STS" is selected (D07_25). The sequence-control-in-progress flag of the presentation status flag "PRDC_STS" is set to ON. In accordance with the result of selection, the *LI-ZHI*-operation-in-progress flag is set to OFF (i.e., nonpresentation is effected), and the transmission-command-edited flag is set to ON (D07_39). The received-command-analysis-in-progress flag is set to OFF (in which a received command is cleared) (D01_3). Processing returns to the main routine.

If the stopping operation is not the first stoppage, a determination is made whether or not the stopping operation is the second stoppage (D07_26). If the stopping operation is the second stoppage, the memory contents in the register (i.e., the type of *LI-ZHI* presentation, and, in this case, the result of a determination as to whether or not *LI-ZHI* presentation is effected) are checked (D07_27). If *LI-ZHI* presentation is not effected, processing pertaining to step (D07_25) and subsequent steps is performed. In contrast, if *LI-ZHI* presentation, processing next to the next step (D07_28). The *LI-ZHI* presentation type flag "RECH_CLS" is checked (D07_29), an address of the sequence control table suited to the *LI-ZHI* type is computed (D07_30).

Next will be described the sequence control table. Fig. 51 is a list for the sequence control table employed in the present embodiment. On the basis of the command transmitted from the main control board 100 in accordance with the statuses of individual games, the sub-control board 200 performs various processing operations. For example, since a token insertion signal has been transmitted from the main control board 100 this time, a token insertion sound code is transmitted to the sound-source IC 206; or, since a command pertaining to the type of presentation to be performed by the main CPU 101 has been transmitted from the main control board 100 this time, *LI-ZHI* presentation or sign presentation is selected, and sound

effects corresponding to the thus-selected presentation is transmitted to the sound-source IC 206. Sound effects are roughly classified into two types. In one type, presentation of sound effects is completed, so long as any one sound output request control code of several sound output request control codes shown in Figs. 58 through 60 is transmitted to the sound-source IC 206 (performance of, e.g., a token insertion sound or a reel spinning start sound, is completed within a comparatively short period of time). In another type, single sound effects are produced by means of transmitting a plurality of sound output request control codes to the sound-source IC 206 under a certain rule (e.g., presentation of single sound effects is performed over a comparatively long period of time, such as a *LI-ZHI* presentation sound and various presentation sounds which would be produced during a bonus game). The sequence control table is provided for effecting presentation of sound effects of latter type. The sequence control table is a data table to be used for selecting a sound output request code and determining the sequence of presentation and the length of sound playback time (*LI-ZHI*-presentation-related sequence control tables are provided in the upper part of Fig. 51, and bonus-game-related sequence control tables are provided in the lower part of the same). For example, Fig. 53 shows a sequence control table to be used when a powerball 3 *LI-ZHI* game is lost. Specifically, when sequence control is commenced, processing operations are sequentially performed; for example, data (mute) provided on the top row are output for a period of 650 ms; super *LI-ZHI* advancement sound (i.e., SD_SPRIN shown in Fig. 59; see the super *LI-ZHI* advancement sound data table shown in Fig. 72 for more detailed sound data) is generated for a period of 167 ms; powerball straining sound is generated for a period of 1683 ms; and a powerball rising sound is generated for a period of time 933 ms. At a point in time when an end code (ENDSQ) has been attained, the sequence control for generating sound effects for losing of a powerball 3 *LI-ZHI* game is completed (see Figs. 54 and 57 for specific examples of another sequence control table).

Turning again to description of the flowchart, a leading address of the sequence control table selected in step (D07_30) is set in the sequence-control-table-access pointer "PRSQPTR" and the sequence-control-table-access pointer backup "SQPTRBK" (D07_31). The sequence-timing-adjustment timer "PR_TIMER" (for managing the sound output time mentioned above) (D07_32) and the presentation status flag of the presentation status flag "PRDC_STS" are set to ON (i.e., such that a *LI-ZHI* presentation is being effected) (D07_33). Further, the sequence-control-in-progress flag of the presentation status flag "PRDC_STS" is set to ON, and the *LI-ZHI*-presentation-in-effect flag is set to ON (such that presentation is effected) (D07_39). The transmission-command-edited flag is also set to ON (D07_39), and the received-command-analysis-in-progress flag is set to OFF (in which a received command is cleared) (D01_3). Processing returns to the main routine.

When NO is selected in step D07_26 (i.e., the third stoppage), a displayed-symbol upgrade flag "RNKUP_FLG" is checked (D07_34), thereby determining whether or not a displayed-symbol upgrade presentation is selected (D07_35). If the presentation is selected, a displayed-symbol upgrade determination sound (BB or RB) sound output code "SD_RKUP" shown in Fig. 60 is set in the register (D07_36). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D07_37), and the presentation status flag of the presentation status flag "PRDC_STS" is set to OFF (i.e., such that *LI-ZHI* presentation is not effected) (D07_38). The sequence-control-in-progress of the presentation status flag "PRDC_STS" is set to ON, and the *LI-ZHI*-presentation-in-effect is set to off (such that presentation is not effected) in accordance with the result of selection (D07_39). The received-command-analysis-in-progress flag is set to OFF (cleared) (D01_3). Processing returns to the main routine.

As mentioned above, even if the player has been notified, over several

consecutive games, of internal generation of winning without the type thereof (e.g., a game in which displayed symbols assume several bonus-winning combinations such as "Do" and "Cake"), the combination which would lead to bonus winning if the player can successfully stop the last reel at a required position is inevitably made clear if the type of bonus winning is forcefully notified on condition that the specified number of games have been played, thus facilitating stopping operation.

<Winning (all-reel-stop) Command Processing: D08>

Figs. 211 through 213 are flowcharts showing procedures for winning (all-reel-stop) command processing.

Figs. 211 through 213 show winning (all-reel-stop) command processing for controlling presentation display and generation of sound effects, which are to be performed by the image display section 13, when all reels have stopped.

In winning (all-reel-stop) command processing, the data stored in the register are first analyzed (D08_1), thereby determining whether or not bonus winning is generated (D08_2). If bonus winning has been generated, bonus winning processing (i.e., processing E02 to be described in detail later) is effected (D08_3), and processing returns to the main routine.

In contrast, if bonus winning has not been generated, the winning flag "WAVEBIT" is checked (D08_4). In accordance with details of generation types "WAVEBIT," inter-bonus-flag check data "FPLY_CHK" are changed (D08_5).

A determination is made as to whether or not small-jackpot winning data are stored in the register (D08_6), as well as to whether or not winning data other than bonus winning data are stored in the register (D08_7). If small-jackpot winning data are not stored in the register and winning data other than bonus winning data are stored in the register, a dropped flag "DROP_FLG" is set (D08_8). In contrast, if small-jackpot winning data are stored in the register (YES is selected in D08_6) or if no small-jackpot winning data are stored in the register (NO is selected in D08_6) or no winning data other than bonus winning data are stored in the register (NO is

selected in D08_7), processing (D08_8) is skipped. Losing of an internally-generated bonus-related winning combination during a normal game is not taken as a drop of winning.

The game status flag "GAMEST" is checked (D08_9), thereby determining whether or not a big bonus game is being played (D08_10). If a big bonus game is being played, there is performed processing pertaining to winning arising during play of a big bonus game (i.e., processing E03 to be described in later) (D08_11). The received-command-analysis-in-progress flag of the presentation status flag is cleared (D01_3). Processing then returns to the main routine.

If a big bonus game is not being played, the game status flag "PRDC_STS" is checked (D08_12), thereby determining whether or not a normal reel screen is displayed (D08_13).

If a normal reel screen is displayed, the small-jackpot winning presentation command "09h" for a normal reel screen is set in the first byte of the transmission-command-edition buffer "TXBUFWK," and a small-jackpot winning presentation instruction is set in the second byte of the same (D08_14).

If a normal reel screen is not displayed, the small-jackpot winning presentation command "0Ah" for an internally-notified-state reel screen is set in the first byte of the transmission-command-edition buffer "TXBUFWK," and the small-jackpot winning presentation instruction is set in the second byte of the same (D08_15).

The stationarily-displayed symbol data 1 "STP_PIC1" are set in the third and fourth bytes of the transmission-command-edition buffer "TXBUFWK" (D08_16), and the displayed-symbol data 1 "DSP_PIC1" are updated (D08_17). Further, the stationarily-displayed symbol data 2 "STP_PIC2" are set in the third and fourth bytes of the transmission-command-edition buffer "TXBUFWK" (D08_18), and the displayed-symbol data 2 "DSP_PIC2" are updated (D08_19).

The presentation status flag "PRDC_STS" is checked (D08_20), thereby

determining whether or not *LI-ZHI* presentation is effected (D08_21) and whether or not *LI-ZHI* winning presentation is effected (D08_22). If both *LI-ZHI* presentation and *LI-ZHI* winning presentation are effected, the presentation status flag "PRDC_STS" is set to an internally-notified-state reel screen (D08_23).

If *LI-ZHI* presentation is not effected (NO is selected in D08_21) or if *LI-ZHI* presentation is effected (YES is selected in D08_21) but *LI-ZHI* winning presentation is not effected (NO is selected in D08_22), processing (D08_23) is skipped.

The transmission-command-edited flag of the presentation status flag "PRDC_STS" is set to ON (D08_24). The data stored in the register are checked (D08_25), thereby determining whether or not winning is generated (D08_26). If no winning is generated, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

If winning is generated, the dropped flag "DROP_FLAG" is checked (D08_27), thereby determining whether or not the winning has been dropped (D08_28) and whether or not the winning is replay winning (D08_29). If the winning has been dropped, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

In contrast, if the winning has not been dropped (NO is selected in D08_28) but is replay winning (YES is selected in D08_29), the replay game sound output code "SD_RPLY" is set in the register (D08_30). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D08_35), the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

Further, if the winning has not been dropped (NO is selected in D08_28) and is not replay winning (NO is selected in D08_29), another determination is made as

to whether or not the winning is winning (effected with payout sound 2) involving payout of 15 gaming tokens shown in Fig. 58 (D08_31).

If the winning is not winning involving payout of 15 gaming tokens, a token payout sound 1 code "SD_PAY1" is selected (D08_32). If the winning involves payout of 15 gaming tokens, a token payout sound 2 "SD_PAY2" is selected (D08_33).

The selected payout sound code is set in the register and the sound backup area 2 "SND_BAK+1" (D08_34), and sound control processing (i.e., processing G01 to be described in detail later) is effected. The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Jackpot Winning Command Processing: D09>

Figs. 214 and 215 are flowcharts showing procedures for jackpot winning command processing.

As shown in Figs. 214 and 215, jackpot winning command processing is for controlling display of presentation and sound effects, which are effected by the image display section 13.

In the jackpot winning command processing, the memory contents in the register are checked (D09_1), and the jackpot winning command "0Eh" is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (D09_2). The bonus-stage-data "BNS_STGN" are checked, and a regular bonus stage number is set in the second byte of the transmission-command-edition buffer "TXBUFWK" (D09_3).

A determination is made as to whether or not the stage of the regular bonus game is stage "3"; that is, whether or not the regular bonus game is in the final stage (D09_4). If the regular bonus game is in the final stage, a sequence-control-table address for the final stage of a regular bonus game shown in Fig. 51 is selected (D09_5), thereby determining whether or not jackpot winning is

in the final round (i.e., the eighth round) (D09_6).

If jackpot winning is in the final round, a sequence-control-table address for the final round of a jackpot game (D09_7) is selected. If jackpot winning is not in the final round, step (D09_7) is skipped.

Next, the number of available regular bonus games "JACGAME" is checked (D09_8), thereby determining whether or not the current game is the final regular bonus game [here, the regular bonus game means a jackpot game (which can be usually played twelve times in one regular bonus round) and does not mean a final regular bonus stage in a regular bonus stage stored in the bonus stage "BNS_STGN"] (D09_9). If the current game is a final round of the regular bonus game, a sequence-control-table address for the final round of a regular bonus game is selected according to the status of the current game; that is, whether or not eight jackpot games provided in stage 3 of the regular bonus shown in Fig. 51 are ended with eight wins, losing-ended with punctures, or winning-ended with punctures (D09_10). If the current game is not a final round of the regular bonus game, step (D09_10) is skipped. If the current game is not in the final stage of the regular bonus game (NO is selected in D09_4), steps (D09_5 through D09_10) are skipped.

The number of jackpot wins is set in the third byte of the transmission-command-edition buffer "TXBUFWK" (D09_11), and the transmission-command-edited flag of the presentation status flag "PRDC_STS" is set to ON (D09_12).

A determination is made as to whether or not jackpot winning has arisen (D09_13). If jackpot winning has arisen, a sequence-control-table address for jackpot winning is selected (D09_14), and a jackpot winning sound code "SD_JAC" shown in Fig. 58 is set in the register (D09_15). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D09_16). If jackpot winning has not arisen (NO is selected in D09_13), steps (D09_14 through D09_16) are skipped.

The selected sequence-control-table address is set in the sequence-control-table-access pointer backup "SQPTRBK" and in the sequence-control-table-access pointer "PRSQPTR" (D09_17). The sequence-timing-adjustment timer "PR_TIMER" is cleared (D09_18), and the sequence-control-in-progress flag of the presentation status flag "PRDC_STS" is set to ON (D09_19). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Payout Completion Command Process: D10>

Fig. 216 is a flowchart showing procedures for payout completion command processing.

As shown in Fig. 216, payout completion command processing is for controlling generation of sound effects, which would be generated when payout of gaming tokens has been completed.

In payout completion command processing, a sound backup area 2 "SND_BAK+1" is cleared (D10_1), a token payout sound mute code "SD_OFF2" is set in the register (D10_2), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D10_3).

A bonus sound backup area "BSND_BK" is checked (D10_4), thereby determining whether or not a sound output request is a bonus sound (start sound) output request (D10_5). If a sound output request is not a bonus sound output request; that is, if payout has not been effected for the winning established by bonus symbols, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

If the sound output is a bonus sound output request, a start sound code corresponding to the type of winning bonus (i.e., any one selected from the BB start sound 1, BB start sound 2, and RB start sound shown in Fig. 58) is set (D10_6), and sound control processing (i.e., processing G01 to be described in detail later) is

effected (D10_7).

A bonus operation sound code corresponding to the type of winning bonus (i.e., either the BB operation sound 1 or RB operation sound 1 shown in Fig. 58) is set in the sound backup area 1 "SND_BAK" (D10_8), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D10_9). The start sound code set in the bonus sound backup area "BSND_BK" is cleared (D10_10), and processing returns to the main loop.

<Bonus-Game-Status Change Instruction Command Processing: D11>

Figs. 217 and 218 are flowcharts showing procedures for bonus-game-status change instruction command processing.

As shown in Figs. 217 and 218, bonus-game-status change instruction command processing is for controlling display of presentation and sound effects, which are effected by the image display section 13, according to the statuses of big bonus and regular bonus games.

In bonus-game-status change instruction command processing, the data stored in the register are checked (D11_1), and a bonus stage display command "0Ch" is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (D11_2).

A determination is made as to whether or not there is a big-bonus end code (D11_3). If there is a big-bonus end code, the bonus stage data "BNS_STGN" are cleared (D11_4). In contrast, if there is not a big-bonus end code, step (D11_4) is skipped.

In accordance with the memory contents in the register, an address corresponding to any one of the bonus stage displays, which are assigned to respective game statuses and are shown in Fig. 39, is selected from the sequence control table (see Fig. 51) (D11_5), thereby checking the presentation status flag "PRDC_STS" (D11_6). A bonus stage type is set in the second byte of the transmission-command-edition buffer "TXBUFWK" (D11_7). The selected

sequence-control-table address is set in the sequence-control-table-access pointer backup "SQPTRBK" and in the sequence-control-table-access pointer "PRSQPTR" (D11_8). The sequence-timing-adjustment timer "PR_TIMER" is cleared (D11_9).

The transmission-command-edited flag of the presentation status flag "PRDC_STS" is set to ON (D11_10), and a bonus operation sound mute code "SD_OFF1" is set (D11_11). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D11_12).

In accordance with the memory contents in the register, a sound code is selected (D11_13), and the thus-selected sound code is set in the sound backup area 2 "SND_BAK+1" (D11_14).

A determination is made as to whether or not a bonus operation sound is being generated (D11_15). If a bonus operation sound is being generated, the selected sound code is set in the register (D11_16). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D11_17). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

If a bonus operation sound is not being generated: that is, if the selected bonus type is the end of a big bonus game or the end of a regular bonus game (NO is selected in D11_15), the stationarily-displayed symbol data 1 "STP_PIC1," the stationarily-displayed symbol data 2 "STP_PIC2," the displayed-symbol data 1 "DSP_PIC1," and the displayed-symbol data 2 "DSP_PIC2" are initialized, thereby setting initial settings for displayed symbols (D11_18). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Operation Command Processing at the End of BB: D12>

Figs. 219 and 220 are flowcharts showing procedures for operation command processing to be effected at the end of a big bonus (called "BB-end operation command processing").

As shown in Figs. 219 and 220, the BB-end operation command processing is for controlling generation of sound effects to be performed at the end of a big bonus game.

In the BB-end operation command processing, the sound backup area "SND_BAK" is checked (D12_1), thereby determining whether or not a payout sound has been generated; namely, whether or not a settlement operation is in progress (D12_2). Put another way, in a case where a settlement operation is currently performed in accordance with a settlement operation command issued previously from the main control board 100 and where the current command is the play-out operation signal shown in Fig. 22, YES is selected.

If settlement operation is in progress, the sound backup area "SND_BAK" is cleared (D12_3), and the token payout sound mute code "SD_OFF2" is set (D12_4). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D12_5), and the memory contents in the register are checked (D12_6). In contrast, if no settlement operation is in progress, steps (D12_3 through D12_6) are skipped.

On the basis of the memory contents in the register, a determination is made as to whether or not a received instruction is a settlement operation instruction (D12_7). If there is a settlement operation instruction, a token payout sound code 1 "SD_PAY1" is set in the sound backup area 2 "SND_BAK+1" and the register, and sound control processing (i.e., processing G01 to be described in detail later) is effected (D12_9). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

In contrast, if there is not a settlement operation instruction (NO is selected in D12_7), a determination is made as to whether or not the received instruction is a play-out instruction (D12_10). If the received instruction is a play-out instruction, the presentation status flag "PRDC_STS" is checked (D12_11), thereby determining

whether or not a sequence control operation is in progress; that is, whether or not big-bonus end presentation is currently being generated (D12_12). If big-bonus end presentation is currently being generated, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

If big-bonus end presentation is not currently being generated, the play-out sound code "SD_OVER" is set in the register and in the sound backup area 1 "SND_BAK" (D12_13), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D12_14). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

If the received instruction is not a play-out instruction (NO is selected in D12_10), the sound backup area 1 "SND_BAK" is checked (D12_15), thereby determining whether or not a play-out sound is currently being generated (D12_16). If a play-out sound is not currently being generated, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

In contrast, if a play-out sound is currently generated, the sound backup area 1 "SND_BAK" is cleared (D12_17), and the sound mute code "SD_OFF1" is set in the register (D12_18). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D12_19). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Error Presentation Command Processing: D13>

Figs. 221 and 222 are flowcharts showing procedures for error presentation command processing.

As shown in Figs. 221 and 222, error command processing is for controlling display of presentation and generation of sound effects, which are performed by the

image display section 13 in the event of occurrence of errors.

In the error command processing, the data stored in the register are first checked (D13_1), thereby determining whether or not error recovery has been performed (D13_2).

If error recovery has not been performed; that is, if errors still remain, an error screen display command "12h" is set in the first byte of the buffer "TXERRWK" specialized for an error screen display command (D13_3). In accordance with the memory contents in the register, the type of error is set in the second byte of the error-screen-display-command-dedicated buffer "TXERRWK" (D13_4). An error sound code "SD_EER" is set in the register and the sound backup area 1 "SND_BAK" (D13_5). The presentation-sequence-timing-adjustment timer "PR_TIMER" is set in a presentation-sequence-control-table-access pointer backup "PRSQTBK" (D13_6). [In other words, in the event that errors have arisen during the course of sequence control processing, information about the length of time in which (or extent to which) the sequence control of the currently-processed sequence control table has been processed is saved. After error recovery, the sequence control processing for the remaining period of time is resumed.] The presentation-sequence-timing-adjustment timer "PR_TIMER" is cleared (D13_7), thereby enabling interrupts (D13_8).

The transmission-command-edited flag of the presentation status flag "PRDC_STS" is set (D13_13), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D13_14). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

If error recovery has been performed (YES is selected in D13_2), the sound backup area 1 "SND_BAK" is cleared (D13_9). An error-status flag of the game status flag "GAMEST" is cleared (D13_10), and an error sound mute code "SD_EROFF" is set in the register (D13_11). The

presentation-sequence-control-table-access pointer backup "PRSQTBK" is set in the presentation-sequence-timing-adjustment timer "PR_TIMER," thereby resetting the timer "PR_TIMER" (D13_12).

The transmission-command-edited flag of the presentation status flag "PRDC_STS" is set (D13_13), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D13_14). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Presentation-Type Command Processing of the Main CPU: D14>

Figs. 223 through 225 are flowcharts showing procedures for presentation-type command processing of the main CPU 101.

As shown in Figs. 223 through 225, presentation-type command processing of the main CPU 101 is for controlling display of presentation and generation of sound effects, which are performed by the image display section 13, in accordance with the status of a game. In more detail, *LI-ZHI* presentation, sign presentation, and sound effects are effected on the basis of the presentation type determined by the main CPU 101, the presentation status, winning type, and gaming status of a sub-CPU 201, and a random number for selecting a presentation.

In the presentation-type command processing of the main CPU 101, a presentation-being-selected flag "PRSELFLG" is checked (D14_1), thereby determining whether or not the presentation-being-selected flag is set to OFF; that is, whether or not a normal game start command has yet to be received (in other words, whether or not processing pertaining to step D04_5 has been performed in the normal game start command processing shown in Fig. 202) (D14_2). If a normal game start command has not yet been received, the received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

In contrast, if the normal game start command has been received, the

presentation status flag "PRDC_STS" is checked (D14_3), thereby determining whether or not sequence control is in progress or whether or not *LI-ZHI* or big bonus ending presentation is currently being performed (D14_4).

If *LI-ZHI* or big bonus ending presentation is currently being performed, a *LI-ZHI*-presentation-in-effect flag and a sequence-control-in-progress of the presentation status flag "PRDC_STS" are cleared (D14_5). *LI-ZHI* presentation sound mute codes "SD_OFF3" and "SD_OFF4" and the big-bonus ending sound mute code "SD_OFF1" are set in the register (D14_6), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D14_7). Conversely, in a case where the sub-control board 200 is performing a control sequence, a normal game start command is transmitted from the main control board 100 only when *LI-ZHI* presentation is being performed or big bonus ending presentation is being performed. The foregoing processing is set for the following reasons: In a case where the main control board 100 manages time, there may sometimes arise a game where the presentation operation being performed by the sub-control board 200 does not end (e.g., lengthy LI-ZHI presentation continues for about 15 seconds) although start of the next game is permitted (e.g., a wait time of 4.1 seconds has elapsed). Many players desire immediate start of the next game. In such a case, presentation processing is forcefully terminated even when the current presentation has not yet ended (i.e., sequence control is in progress), thus enabling starting of the next game.

If *LI-ZHI* or big bonus ending presentation is not being performed, steps (D14_5 through D14_7) are skipped.

Subsequently, presentation selection processing (i.e., processing E1 to be described in detail) is performed (D14_8), thus checking the reel screen type of the presentation status flag "PRDC_STS" (D14_9) and determining whether or not a normal reel screen is being displayed (D14_10).

If a normal reel screen is not displayed, a spinning start command "05h" on

an internally-notified-state reel screen is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (D14_11). In contrast, if a normal reel screen is being displayed, a spinning start command "04h" on a normal reel screen is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (D14_12).

The sign presentation type "PRE_CLS" is checked, and sign presentation is set in second byte of the transmission-command-edition buffer "TXBUFWK" (D14_13). The displayed-symbol data 1 "DSP_PIC1" and the displayed-symbol data 2 "DSP_PIC2" are set in the third and fourth bytes of the transmission-command-edition buffer "TXBUFWK" (D14_14).

A determination is made as to whether or not a sign presentation greater than or equal to a Yah-Hoo sign presentation code is instructed (D14_15). If a sign presentation [i.e., a Yah-Hoo sign or (BB or RB) bonus determination presentation] greater than or equal to a Yah-Hoo sign presentation code is not instructed, the game start sound code (start sound 1 or 2) shown in Fig. 58 is set (D14_27). Sound control processing (i.e., processing G01 to be described in detail later) is effected (D14_28), and the presentation-being-selected flag "PRSELFLG" is cleared (D14_29). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

If sign presentation greater than or equal to a Yah-Hoo sign presentation code (having a high probability of bonus winning) is instructed, a determination is made as to whether or not the instructed presentation is a bonus determination presentation (balloon *LI-ZHI*) (D14_16). If the instructed presentation is not a bonus determination presentation, a sign sound code "SD_PRE" (start sound 3) shown in Fig. 60 is set in the register (D14_25), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D14_26). The presentation-being-selected flag "PRSELFLG" is cleared (D14_29). The

received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

In contrast, if the instructed presentation is a bonus determination presentation, a *LI-ZHI* presentation flag of the presentation status flag "PRDC_STS" is set to ON; the sequence-control-in-progress flag of the same is set to ON; and a reel-screen type flag of the same is also set to ON (D14_17). The winning flag "WAVEBIT" is checked (D14_18), thereby determining whether or not regular bonus winning has arisen (D14_19).

If regular bonus winning has not arisen, an address in a BB-determination-balloon-*LI-ZHI*-sequence control table is set in the sequence-control-table-access pointer "PRSQPTR" and the sequence-control-table-access pointer backup "SQPTRBK" (D14_20). A sign presentation code to be transmitted is taken as big-bonus determination and reset in the second byte of the transmission-command-edition buffer "TXBUFWK" (D14_21).

If regular bonus winning has arisen, the address in a BB-determination-balloon-*LI-ZHI*-sequence control table is set in the sequence-control-table-access pointer "PRSQPTR" and the sequence-control-table-access pointer backup "SQPTRBK" (D14_22). The sign presentation code to be transmitted is taken as regular bonus determination and reset in the second byte of the transmission-command-edition buffer "TXBUFWK" (D14_23).

The presentation-sequence-timing-adjustment timer "PR_TIMER" is cleared (D14_24), and the presentation-being-selected flag "PRSELFLG" is cleared (D14_29). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Sound Single Command Processing: D15_

Fig. 226 is a flowchart showing processing for sound single command

processing.

As shown in Fig. 226, sound single command processing corresponds to sound output control processing for sound outputs, in which a real sound type (on the level of the sound output request control codes shown in Figs. 58 through 60 in the present embodiment) is determined by means of the selection processing in the main control board 100, without involvement of presentation selection processing performed by the sub-control board 200. Further, sound single command processing is for backing up sound data requiring backup.

In sound single command processing, the data stored in the register are first checked (D15_1), thereby determining whether or not payout sound is specified (D15_2). If a payout sound has been specified, a payout sound code is set in the sound backup area 2 "SND_BAK+1" (D15_3). If a payout sound has not been specified, step (D15_3) is skipped.

A sound code is set in the register (D15_4), and sound control processing (i.e., processing G01 to be described in detail later) is effected (D15_5). The received-command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is cleared (D01_3), and processing returns to the main routine.

<Presentation Selection Processing During Normal Game and Internally-Generated Bonus Game: E01>

Figs. 227 through 230 are flowcharts showing procedures for presentation selection processing to be effected during a normal game and an internally-generated bonus game.

As shown in Figs. 227 through 230, this processing is for controlling presentation display performed by the image display section 13 during a normal game and an internally-generated bonus game, in accordance with the status of the game.

In this processing, the inter-bonus-flag check data "FPLY_CHK" (E01_1) are checked, and the displayed-symbol upgrade flag "RNKUP_FLG" is checked (E01_2).

The presentation status flag "PRDC_STS" is checked (E01_3), thereby determining whether or not presentation is currently being provided on the internally-notified-state reel screen (E01_4). If presentation is currently being provided on the internally-notified-state reel screen, the counter "WPLY_CNT" for counting the number of games after illumination of the WIN lamps is checked (E01_5), thereby determining whether or not the displayed-symbol data have already been changed (E01_6) and whether or not less than 10 games remain after winning presentation has been performed (E01_7).

Here, the expression "whether the displayed-symbol data have already been changed" signifies whether or not a displayed-symbol upgrade flag has been set in a past game and whether the bonus type of the displayed symbols has been determined and displayed.

If the displayed-symbol data have not been changed and ten or more games still remain after winning presentation has been performed, the winning flag "WAVEBIT" is checked (E01_8), thereby determining whether or not a big bonus has been generated (E01_9). If a big bonus has been generated, "7," which is a symbol to be displayed on the left wheel when a big bonus game is generated, is set in the stationarily-displayed symbol data 1 "STP_PIC1." Further, "7," which is a symbol to be displayed on the right and center reels when a big bonus game is generated, is set in the stationarily-displayed symbol data 2 "STP_PIC2" (E01_10). An upgrade flag "RKUP_BBDT" for ranking displayed symbols as BB-determination displayed symbols is selected (E01_11).

The selected upgrade flag is set in the displayed-symbol upgrade flag "RNKUP_FLG" (E01_14). Further, a table "BNFGPLAY" for selecting a winning sign during internal generation of a bonus shown in Fig. 97 is selected (E01_15). The bonus generation bit of the winning flag "WAVEBIT" is masked (i.e., small-jackpot presentation is prioritized) and set in the register (E01_16).

Subsequently, a determination is made as to whether or not small-jackpot

winning is generated (E01_17). If no small-jackpot winning is generated, a winning flag is rechecked and set in the register (E01_18). Winning sign presentation type selection processing (i.e., processing F05 to be described in detail later) is effected (E01_19), and processing returns to the main routine. In contrast, if small-jackpot winning is generated, step (E01_18) is skipped.

If the displayed symbol data have already been changed (YES is selected in E01_6) and if less than ten games remain after winning presentation has been performed (YES is selected in E01_7), the table "BNFGPLAY" is selected (E01_15). The bonus generation bit of the winning flag "WAVEBIT" is masked (i.e., small-jackpot presentation is prioritized) and set in the register (E01_16).

Subsequently, a determination is made as to whether or not small-jackpot winning is generated (E01_17). If no small-jackpot winning is generated, a winning flag is rechecked and set in the register (E01_18). The winning sign presentation type selection processing (i.e., processing F05 to be described in detail later) is effected (E01_19), and processing returns to the main routine. In contrast, if small-jackpot winning is generated, step (E01_18) is skipped.

If not big bonus has been generated (NO is selected in E01_9), "BAR," which is a symbol to be displayed on the left reel when a regular bonus is generated, is set in the stationarily-displayed symbol data 1 "STP-PIC1", and then "BAR," which is a symbol to be displayed on the right and center reels when a regular bonus is generated, is set in the stationarily-displayed symbol data 2 "STP_PIC2" (E01_12). The upgrade flag "RKUP_BBDT" for ranking displayed symbols as BB-determination displayed symbols is selected (E01_13). The selected upgrade flag is set in the displayed-symbol upgrade flag "RNKUP_FLG" (E01_14). Further, a table "BNFGPLAY" for selecting a winning sign during internal generation of a bonus shown in Fig. 97 is selected (E01_15). The bonus generation bit of the winning flag "WAVEBIT" is masked (i.e., small-jackpot presentation is prioritized) and set in the register (E01_16).

Subsequently, a determination is made as to whether or not small-jackpot winning is generated (E01_17). If no small-jackpot winning is generated, a winning flag is rechecked and set in the register (E01_18). The winning sign presentation type selection processing (i.e., processing F05 to be described in detail later) is effected (E01_19), and processing returns to the main routine. In contrast, if small-jackpot winning is generated, step (E01_18) is skipped.

If no presentation is performed on the internally-notified-state reel screen (NO is selected in E01_4), the game status flag "GAMEST" is checked (E01_20), thereby determining whether or not internal generation of a winning mode is currently notified; that is, whether or not the WIN lamps are illuminated (E01_21).

If the WIN lamps are not illuminated, the inter-bonus-flag check data "FPLY_CHK" (E01_22) are checked, thus determining whether or not a bonus game is internally generated (E01_23). If no bonus game is internally generated, a table GNRLRECH for selecting *LI-ZHI* during a normal game shown in Figs. 77 through 88 is selected (E01_24). If a bonus game is internally generated, a table "BNFGRECH" for selecting *LI-ZHI* during internal generation of a bonus game shown in Figs. 82 through 86 is selected (E01_26). If the WIN lamps are illuminated (YES is selected in E01_21), a table "WLONRECH" for selecting *LI-ZHI* during illumination of the WIN lamps shown in Figs. 87 through 90 is selected (E01_25).

LI-ZHI presentation type selection processing (i.e., processing F06 to be described in detail later) is effected (E01_27). *LI-ZHI* presentation type "RECH_CLS" is checked (E01_28), thereby determining whether or not *LI-ZHI* presentation is performed (E01_29). If *LI-ZHI* presentation is effected, symbols to be displayed have already been selected by means of selection of *LI-ZHI* presentation, and processing returns to the main routine.

If *LI-ZHI* presentation is not effected, a table "GNRLPLAY" for selecting a winning sign during a normal game is selected (E01_30). The inter-bonus-flag

check data "FPLY_CHK" (E01_31) are checked, thus determining whether or not a bonus game is internally generated (E01_32).

If a bonus game is internally generated, the bonus generation bit of the winning flag "WAVEBIT" is masked (i.e., small-jackpot presentation is prioritized) and set in the register (E01_33). A determination is made as to whether or not small-jackpot winning is generated (E01_34). If no small-jackpot winning is generated, the winning flag is rechecked and set in the register (E01_35). In contrast, if small-jackpot winning is generated, step (E01_35) is skipped.

A table "BNFGPLAY" for selecting a winning sign during internal generation of a big bonus is selected (E01_36). Winning sign presentation type selection processing (i.e., processing F05 to be described in detail later) is effected (E01_37), and processing returns to the main routine.

If a bonus game is not internally generated (NO is selected in E01_32), steps (E01_33 through E01_36) are skipped. The winning sign presentation type selection processing (i.e., processing F05 to be described in detail later) is effected (E01_37), and processing returns to the main routine.

<Bonus Winning Processing: E02>

Figs. 231 and 232 are flowcharts showing procedures for bonus winning processing.

As shown in Figs. 231 and 232, bonus winning processing is for controlling presentation display and sound effects, which are effected by the image display section 13, in accordance with the status of a game at the time of generation of a bonus.

In bonus winning processing, a winning bonus type is determined by means of analysis of the data stored in the register (E02_1). A bonus winning presentation command "0Bh" is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (E02_2). A winning bonus type is set in the second byte of the transmission-command-edition buffer "TXBUFWK" (E02_3). The

transmission-command-edited flag of the presentation status flag "PRDC_STS" is set to ON, and the reel screen type flag of the same is set to OFF, thus indicting that a normal screen is displayed (E02_4).

The presentation status flag "PRDC_STS" is checked (E02_5), thereby determining whether or not *LI-ZHI* presentation is effected (E02_6). If *LI-ZHI* presentation is effected, the *LI-ZHI* presentation flag and the sequence-control-in-progress flag of the presentation status flag "PRDC_STS" are set to OFF (E02_7). The *LI-ZHI* presentation sound mute codes "SD_OFF3" and "SD_OFF4" are set in the register (E02_8), and sound control processing (i.e., processing G01 to be described in detail later) is effected (E02_9). If *LI-ZHI* presentation is not effected (NO is selected in E02_6), steps (E02_7 through E02_9) are skipped. This processing means that, when big bonus winning, for example, is generated as a result of a player having performed stopping operation while *LI-ZHI* presentation is effected by the sub-control board 200, the *LI-ZHI* presentation is terminated, thereby effecting winning presentation.

By means of such a configuration, the player does not experience an unfavorable sensation, which would otherwise be induced when presentation is continued even though winning has already been generated (i.e., Jackpot winning or losing has been determined).

A token payout sound code 2 "SD_PAY2" is set in the register and the sound backup area 2 "SND_BAK+1" (E02_10), and sound control processing (i.e., processing G01 to be described in detail later) is effected (E02_11).

A determination is made as to whether or not regular bonus winning has been generated (E02_12). If regular bonus winning has been generated, a regular bonus start sound code "SD_RBHIT" and a regular bonus operation sound 1 code "SD_RBBGM1" are set (E02_13). The type of bonus stage is set as RB"00" (E02_14).

The selected bonus stage type is set in a bonus stage area "BNS_STGN"

(E02_19), and the selected sound code is set in the register and a bonus sound backup area "BSND_BK" (E02_20). The inter-bonus-flag check data "FPLY_CHK" are cleared (E01_21), and processing returns to the main routine.

If the winning is not regular bonus winning (NO is selected in E02_12), a determination is made as to whether or not big bonus winning has been made by "White 7" or "Red 7" (E02_15).

If big bonus winning has been made by "Red 7," a big bonus start sound 2 code "SD_BBHIT2" and a big bonus operation sound 2 code "SD_BBBGM2" are set (E02_16).

If big bonus winning has been made by "White 7," a big bonus start sound 1 code "SD_BBHIT1" and a big bonus operation sound 1 code "SD_BBBGM1" are set (E02_17).

Subsequently, the type of bonus stage is set to BB stage 1 start "02" (E02_18), and the selected bonus stage type is set in the bonus stage area "BNS_STGN" (E02_19). The selected sound code is set in the register and the bonus sound backup area "BSND_BK" (E02_20). The inter-bonus-flag check data "FPLY_CHK" are cleared (E01_21), and processing returns to the main routine.

<Winning Processing to be Effected during a Big Bonus Game: E03>

Figs. 233 through 236 are flowcharts showing procedures for winning processing to be effected during a big bonus game.

As shown in Figs. 233 through 236, this processing is for controlling presentation display and sound effects, which are effected by the image display section 13, in accordance with the status of a game at the time of big bonus winning.

In this processing, the type of small-jackpot winning is determined by means of analysis of the data stored in the register (E03_1), thereby determining whether or not the small-jackpot winning is regular bonus winning (E03_2).

If the small-jackpot winning is not the regular bonus winning generated during a big bonus game, a determination is made as to whether or not a mismatch

exists between the type of generated winning and the type of winning; that is, whether or not the data stored in the second byte of a winning type command, shown in Fig. 18 and transmitted from the main control board 100, represent 0 (losing) in spite of the fact that there has been transmitted a game start command for a normal game, shown in Fig. 14, in which data stored in the second byte of the command represent internal generation of winning (E03_3). If a mismatch exists between the type of generated winning and the type of winning, the dropped flag "DROP_FLG" is set to ON (E03_4). In contrast, if no mismatch exists (i.e., if a match exists) between the type of generated winning and the type of winning, step (E03_4) is skipped.

A command code "DSP_BNHIT(10h)" pertaining to small-jackpot winning generated in a regular bonus game during a big bonus game is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (E03_21). The bonus stage area "BNS_STGN" is checked, and a big-bonus stage number is set in the third byte of the transmission-command-edition buffer "TXBUFWK" (E03_22). The number of remaining big bonus games "BBPCTR" is checked and is set in the third byte of the transmission-command-edition buffer "TXBUFWK" (E03_23). The memory contents in the register (i.e., the type of small-jackpot winning) are checked, and the type of small-jackpot winning is set in the fourth byte of the transmission-command-edition buffer "TXBUFWK" (E03_24). The transmission-command-edited flag of the presentation status flag "PRDC_STS" is set to ON (E03_25).

A determination is made as to whether or not winning is generated (E03_26) and, if winning has been generated, whether or not the winning has been dropped (E03_37). If no winning is generated or winning has been dropped, processing returns to the main routine.

If winning is generated and no winning has been dropped, a determination is further made as to whether or not the winning involves payout sound 2 or whether

or not the winning involves payout of 15 gaming tokens (E03_28). If the winning does not involve payout of 15 gaming tokens, the payout sound 1 code "SD_PAY1" shown in Fig. 58 is selected (E03_29). In contrast, if the winning involves payout of fifteen gaming tokens, the payout sound 2 code "SD_PAY2" is selected (E03_30).

The thus-selected payout sound is set in the register and the sound backup area 2 "SND_BAK+1" (E03_31), and sound control processing (i.e., processing G01 to be described in detail later) is effected (E03_32). A token get sound code "SD_CGET" shown in Fig. 60 is set in the register (E03_33), and sound control processing (i.e., processing G01 to be described in detail later) is effected (E03_34). Processing returns to the main routine.

When regular bonus winning has arisen in the big bonus game (YES is selected in E03_2), a presentation code "11h" pertaining to regular bonus winning arising during a big bonus game is set in the first byte of the transmission-command-edition buffer "TXBUFWK" (E03_5). The bonus stage area "BNS_STGN" is checked, and a regular-bonus stage number is set in the second byte of the transmission-command-edition buffer "TXBUFWK" (E03_6). The transmission-command-edited flag of the presentation status flag "PRDC_STS" is set to ON (E03_7).

A regular-bonus start sound code "SD_JACIN1" shown in Fig. 58 is stored in the register and the bonus sound backup area "BSND_BK" (E03_8). Sound control processing (i.e., processing G01 to be described in detail later) is effected (E03_9).

The token payout sound 1 code "SD_PAY1" is set in the sound backup area 2 "SND_BAK+1" (E03_10), and sound control processing (i.e., processing G01 to be described in detail later) is effected (E03_11). The bonus stage area "BNS_STGN" is checked (E03_12), thereby determining whether or not the regular bonus game is in stage 3; that is, whether or not the regular bonus game is in the final stage (E03_13).

If the regular bonus game is in the final stage, the number of remaining big

bonus games "BBPCTR" is checked (E03_14), thereby determining whether or not restoration from puncture has been effected; that is, whether or not regular bonus winning has been achieved in the final big bonus game (E03_15).

If regular bonus winning has been achieved in the final big bonus game, a restoration-from-punctures (failures) sound code "SD_PRET" shown in Fig. 60 is set (E03_16), and sound control processing (i.e., processing G01 to be described in detail later) is effected (E03_17). In contrast, if regular bonus winning has not been achieved in the final big bonus game, steps (E03_16 and E03_17) are skipped.

A regular bonus operation sound 2 code "SD_RBBGM2" shown in Fig. 58 is selected (E03_18), and the thus-selected regular bonus operation sound code is set in the bonus sound backup area "BSND_BK" (E03_19). Processing returns to the main routine.

If the current stage is not the final stage of the regular bonus (NO is selected in E03_13), a regular bonus operation sound 1 code "SD_RBBGM1" is selected (E03_20). The thus-selected regular-bonus operation sound code is set in the bonus sound backup area "BSND_BK" (E03_19). Processing returns to the main routine.

As mentioned above, in a case where regular bonus winning has arisen in the final normal game during the big bonus, special sound presentation is effected. Accordingly, clearing of all normal games during the big bonus can be announced not only to the player but also to other players around him, thereby increasing the entertainment value of a game.

<Presentation Status Reset Processing: F01>

Fig. 237 is a flowchart showing procedures for presentation status reset processing.

As shown in Fig. 237, status reset processing is for initializing presentation display and generation of sound effects, which are to be performed by the image display section 13.

In presentation status reset processing, a sound initialization code "SDRESET" shown in Fig. 58 is set in the register (F01_1), and sound control processing (i.e., processing G01 to be described in detail later) is effected (F01_2). A liquid-crystal display erasure (initialization) command "01h" is set in the transmission-command-edition buffer "TXBUFWK" (F01_3). The transmission-command-edited flag of the presentation status flag "PRDC_STS" is set to ON (F01_4). Processing returns to the main routine.

<Sound Restoration Processing: F02>

Fig. 238 is a flowchart showing procedures for sound restoration processing.

As shown in Fig. 238, sound restoration processing is for reconstructing backup sound data.

In sound restoration processing, the sound backup area 1 "SND_BAK" in which are stored sound data to be played back on CH1 of the sound-source IC 206 is checked (F02_1), thereby determining whether or not backup data are stored in the backup area (F02_2). If data are backed up in the area, the backed-up sound control request code is set in the register (F02_3), and sound control processing (i.e., processing G01 to be described in detail later) is effected (F02_4). If no data are backed up, steps (F02_3 and F02_4) are skipped.

Next, the sound backup area 2 "SND_BAK+1" in which are stored sound data to be played back on CH2 of the sound-source IC 206 is checked (F02_5), thereby determining whether or not backup data are stored in the backup area (F02_6). If data are backed up in the area, the backed-up sound control request code is set in the register (F02_7), and sound control processing (i.e., processing G01 to be described in detail later) is effected (F02_8). If no data are backed up, steps (F02_7 and F02_8) are skipped.

The sound backup area 3 "SND_BAK+2" in which are stored sound data to be played back on CH3 and CH4 of the sound-source IC 206 is checked (F02_9), thereby determining whether or not backup data are stored in the backup area

(F02_10). If data are backed up in the area, the backed-up sound control request code is set in the register (F02_11), and sound control processing (i.e., processing G01 to be described in detail later) is effected (F02_12). If no data are backed up, steps (F02_11 and F02_12) are skipped.

<Received Command Storage Processing: F03>

Fig. 239 is a flowchart showing procedures for received-command storage processing.

As shown in Fig. 239, received-command storage processing is for storing data transmitted from the main control board 100.

In received command storage processing, data are extracted from a received-command higher-byte area "RCVCMDH" and a received-command lower-byte area "RCVCMDL." The thus-extracted data are stored in a received-data area "RXBUFF" (F03_1). The number of registered commands is updated, and the thus-updated number is stored in a number-of-received-data area "RECCNT" (F03_2). Processing returns to the main routine.

<Command Fetch Processing: F04>

Fig. 240 is a flowchart showing procedures for command fetch processing.

As shown in Fig. 240, command fetch processing is for analyzing commands.

In command fetch processing, a received command is extracted from the received-data area "RXBUFF," and the thus-extracted command is stored in an upper-byte-area-of-command-being-analyzed "ALCMD_HI" and in a lower-byte-area-of-command-being-analyzed "ALCMD_LO" (F04_1). The command-being-analyzed "ALCMD_HI" and the command-being-analyzed "ALCMD_LO" are set in the register (F04_2).

Next, "1" is subtracted from the number of received data sets "RECCNT," thereby updating the number of commands registered in the buffer (F04_3). The command-analysis-in-progress flag of the presentation status flag "PRDC_STS" is set to ON (F04_4). Interrupts are enabled (F04_5), and processing returns to the

main routine.

<Processing for Selecting the Type of Winning Sign Presentation: F05>

Figs. 241 and 242 are flowcharts showing procedures about processing for selecting the type of winning sign.

As shown in Figs. 241 and 242, this processing is for selecting the type of winning sign presentation and presentation display to be performed by the image display section 13 (specifically displayed symbols).

In this processing, a random number for selection purpose is extracted from presentation selection random numbers "SELRAND" (random numbers ranging from 0 to 65535) (F05_1). Sign presentation is determined from the thus-selected winning sign presentation selection table and the register (a winning flag), and the thus-selected sign is set in the sign presentation type "PRE_CLS" (F05_2).

The *LI-ZHI* presentation type "RECH_CLS" is cleared (F05_3), and the presentation status flag "PRDC_STS" is checked (F05_4), thereby determining whether or not an internally-notified-state reel screen is displayed; that is, whether or not there is a necessity for selecting symbols to be displayed (F05_5). In a case where an internally-notified-state reel screen display is displayed, there is no necessity for selecting symbols to be displayed. Hence, processing returns to the main routine.

In contrast, if the internally-notified-state reel screen is not displayed, a displayed-symbol section table number is set in a displayed-symbol selection table number "DEMEDATA" by reference to the sign presentation table selected by way of processing E01 (F05_6).

A random number is extracted from the presentation selection random numbers "SELRAND" (F05_7). A center displayed symbol is selected on the basis of the thus-selected displayed-symbol selection table and random number (F05_8). The selected center-displayed-symbol data are set in a center-displayed-symbol data save area "SREEL_BK" (F05_9).

Another random number is extracted from the presentation selection random numbers "SELRAND" (F05_10). A right-side displayed symbol is selected on the basis of the thus-selected displayed-symbol selection table and random number (F05_11). The selected right-side-displayed-symbol data are set in a right-side-displayed-symbol data save area "RREEL_BK" (F05_12). The right-side-displayed-symbol data and the center-displayed-symbol data are merged into a single data, and the thus-merged data are set in the stationarily-displayed symbol data 2 "STP_PIC2" (F05_13).

Further, another random number is extracted from the presentation selection random numbers "SELRAND" (F05_14). A symbol to be displayed on the left-side reel is selected on the basis of the selected displayed-symbol selection table and random number (F05_15). A determination is made as to whether or not the selected displayed symbols constitute a bonus combination of symbols (F05_16).

If the thus-selected displayed-symbols constitute a bonus combination of symbols, the center-displayed-symbol data save area "SREEL_BK" is checked (F05_17), thus determining whether or not the selected displayed-symbol image is identical with the center displayed symbol (F05_18).

If the selected displayed symbol is identical with the center displayed symbol, the right-side displayed-symbol data (RREEL_BK) is checked (F05_19), thus determining whether or not the selected displayed-symbol image is identical with the right-side displayed symbol (F05_20).

If the selected displayed symbol is identical with the right-side displayed symbol, the selected displayed symbol is shifted downward by 1 frame (F05_21). The selected displayed-symbol is set in the stationarily-displayed symbol data 1 "STP_PIC1" (F05_22). Processing returns to the main routine. By means of these processing operations, the displayed symbols match the symbols of the generated winning mode, thus preventing occurrence of a mismatch between the displayed symbols and the status of the game.

If the selected displayed symbol differs from a bonus symbol (NO is selected in F05_16), step (F05_17) is skipped. If the selected displayed symbol differs from the center displayed-symbol (NO is selected in F05_18), step (F05_19) is skipped. If the selected displayed symbol differs from the right-side displayed-symbol (NO is selected in F05_20), step (F05_21) is skipped. The selected displayed-symbol is set in the stationarily-displayed symbol data 1 "STP_PIC1" (F05_22). Processing returns to the main routine.

<*LI-ZHI* Presentation-Type Selection Processing: F06>

FIGS. 243 through 245 are flowcharts showing procedures for *LI-ZHI* presentation-type selection processing.

As shown in Figs. 243 through 245, *LI-ZHI* presentation-type selection processing is for controlling presentation display to be performed by the image display section 13.

In *LI-ZHI* presentation-type selection processing, a random number for selection purpose is extracted from presentation selection random numbers "SELRAND" (F06_1). On the basis of the *LI-ZHI* presentation selection table selected in processing E01 and the extracted random number, the type of *LI-ZHI* presentation and the type of *LI-ZHI* sign presentation are determined (F06_2). In the case where *LI-ZHI* presentation is to be effected, selection of symbols to be displayed is not performed.

The selected type of *LI-ZHI* presentation is set in *LI-ZHI* presentation type "RECH_CLS" (F06_3), thus determining whether or not *LI-ZHI* sign presentation is to be effected (F06_4). If no *LI-ZHI* sign presentation is to be effected, processing returns to the main routine.

If *LI-ZHI* sign presentation is to be effected, the selected sign presentation type is set in the sign presentation type "PRE_CLS" (F06_5), thereby determining whether or not *LI-ZHI* winning has arisen (F06_6).

If *LI-ZHI* winning has arisen, the winning flag "WAVEBIT" is checked

(F06_7), thereby checking whether or not big bonus winning is generated (F06_8). If big bonus winning is not generated, a table "RBRECHDAT" for selecting displayed symbols at the time of regular bonus winning presentation shown in Fig. 93 is selected (F06_9). If big bonus winning is generated, a table "BBRECHDAT" for selecting displayed symbols at the time of big bonus winning presentation shown in Fig. 92 is selected (F06_11).

If no *LI-ZHI* winning has arisen (NO is selected in F06_6), a table "MSRECHDAT" for selecting displayed symbols at the time of *LI-ZHI* losing presentation shown in Fig. 94 is selected (F06_10).

A random number for selection purpose is extracted from presentation selection random numbers "SELRAND" (F06_12). On the basis of the extracted random number and the selected table for selecting symbols to be displayed at the time of *LI-ZHI* winning, symbols to be displayed (i.e., *TEN P'AIS* symbols) are determined (F06_13).

Subsequently, a determination is made as to whether or not *LI-ZHI* losing has arisen (F06_14). If *LI-ZHI* losing has not arisen, a determination is made as to whether or not *TEN P'AIS* symbols correspond to bonus-type determination symbols ("7" or "BAR") (F06_15). If *TEN P'AIS* symbols correspond to bonus-type determination symbols, the displayed-symbol change counter "WPLY_CNT" is cleared (F06_16). In contrast, if *TEN P'AIS* symbols do not correspond to bonus-type determination symbols, the initial value "PCHG_NM" is set in the displayed-symbol change counter "WPLY_CNT" (F06_17).

If *LI-ZHI* losing has arisen (YES is selected in F06_14), steps (F06_15 through F06_17) are skipped.

Subsequently, a determination is made as to whether or not balancing-on-rolling-ball *LI-ZHI* has arisen (F06_18). If balancing-on-rolling-ball *LI-ZHI* has arisen, a center displayed symbol is determined from the table for selecting a center symbol to be displayed at the time of balancing-on-rolling-ball

LI-ZHI losing (F06_23). The right-side-displayed-symbol data and the center-displayed-symbol data are merged into a single data, and the resultant data are set in the stationarily-displayed symbol data 2 "STP_PIC2" (F06_24). The left-side-displayed-symbol data are set in the stationarily-displayed symbol data 1 "STP_PIC1" (F06_25). Processing returns to the main routine.

If no balancing-on-rolling-ball *LI-ZHI* has arisen (NO is selected in F06_18), a random number for selection purpose is extracted from presentation selection random numbers "SELRAND" (F06_19). On the basis of the extracted random number and the selected table "MSRECHDAT," a center symbol to be displayed is determined (F06_20).

A determination is made as to whether or not all the displayed symbols; i.e., the left-side displayed symbol, the center displayed symbol, and the right-side displayed symbol, are identical with one another (F06_21). If they are identical, the center displayed symbol is shifted upward by 1 frame (F06_22). By means of these processing operations, the displayed symbols come to match the symbols of the generated winning mode, thus preventing occurrence of a mismatch between the displayed symbols and the status of the game. If not all of the three displayed symbols are identical, step (F06_22) is skipped.

The right-side-displayed-symbol data and the center-displayed-symbol data are merged into a single data, and the resultant data are set in the stationarily-displayed symbol data 2 "STP_PIC2," (F06_24) and the left-side-displayed-symbol data are set in the stationarily-displayed symbol data 1 "STP_PIC1" (F06_25). Processing returns to the main routine.

<Processing for Selecting Displayed Symbol from Selection Table: F07>

Fig. 246 is a flowchart showing procedures for processing for selecting a displayed symbol from a selection table.

As shown in Fig. 246, this processing is for controlling presentation display to be performed by the image display section 13, by means of selecting symbols to

be displayed from a displayed-symbol selection table.

In this processing, a random number for selection purpose is extracted from presentation selection random numbers "SELRAND" (F07_1). On the basis of the extracted random number and the displayed-symbol selection table number "DEMEDATA," symbols to be displayed are selected (F07_2). Processing returns to the main routine.

<Sound Control Processing: G01>

Fig. 247 is a flowchart showing procedures for sound control processing.

As shown in Fig. 247, sound control processing is for controlling generation of sound effects by means of checking a sound request control code.

In sound control processing, a sound request control code stored in the register is first checked (G01_1), thus determining whether or not there is a sound control request (G01_2). If there is no sound control request, processing returns to the main routine.

In contrast, if there is a sound control request, a determination is made as to whether or not the sound control request is an initialization code (G01_3) and, if the sound request is not an initialization code, a determination is made as to whether or not the request is a mute request code (G01_5).

If the sound control request is an initialization code, sound initialization processing (i.e., processing G02 to be described in detail later) is effected (G01_4). Then, processing returns to the main routine (see Fig. 58). If the sound control request is a mute request code, sound mute processing (i.e., processing G03 to be described in detail later) is effected (G01_6). Then, processing returns to the main routine (see Fig. 58). If the sound control request corresponds to neither a sound initialization request code nor a sound mute request code, sound output processing (i.e., processing G4 to be described in detail later) is effected (G01_7). Then, processing returns to the main routine (see Fig. 58).

<Sound Initialization Processing: G02>

Fig. 248 is a flowchart showing procedures for sound initialization processing.

As shown in Fig. 248, sound initialization processing is for initializing sound effects to be generated.

In this processing, there is performed all-channel-playback-stop processing (i.e., processing G06 to be described in detail later) (G02_1), and all channel reset data are stored in the register (G02_2).

Subsequently, SD_OUT sound output data transmission processing (i.e., processing G05 to be described in detail later) is effected (G02_3), and a restoration-effective-sound-output-status storage area "PLAY_NUM" is cleared (G02_4). A prioritized-single-sound-output-status storage area "HIT_NUM" is cleared (G02_5). Processing returns to the main routine.

<Sound Mute Processing: G03>

Figs. 249 and 250 are flowcharts showing procedures for sound mute processing.

As shown in Figs. 249 and 250, this processing is for muting sound effects.

In the processing, a determination is made as to whether or not the sound control request is an error sound mute code "RESUME" shown in Fig. 76 (G03_1). If the sound control request is not an error sound mute code "RESUME", the restoration-effective-sound-output-status storage area "PLAY_NUM" is cleared (G03_2). A playback stop command code "CMD_QUIT" for specifying a channel on which playback is to be stopped is set (G03_3). SD_OUT sound output data transmission processing (i.e., processing G05 to be described in detail later) is performed (G03_4). Processing returns to the main routine.

If the sound control request is the error sound mute code "RESUME", ALL_OFF error sound mute processing is performed (G03_5). Data pertaining to the specified channel are loaded from the restoration-effective-sound-output-status storage area "PLAY_NUM" to the register (G03_6).

A determination is made as to whether or not sound output request data are stored in the register (G03_7). If sound output request data are stored in the register, sound output processing (sound processing pertaining to a predetermined sound number) is performed. In contrast, if sound output request data are not stored in the register, the address is changed to the next channel (G03_8). After all the channels have been subjected to the above-described processing operations (G03_6 through G03_8) (G03_9), processing returns to the main routine.

<Sound Output Processing: G04>

Figs. 251 through 254 are flowcharts showing procedures for sound output processing.

As shown in Figs. 251 through 254, this processing is for producing corresponding sound effects in accordance with the types of sound output data set in the register.

In the processing, the data stored in the register are compared with the total number of sounds "SDDT" shown in Fig. 76 (G04_1), thereby determining whether or not the number of the sounds stored in the register is greater than the total number of sounds (G04_2). If the number of sounds is greater than the total number, processing returns to the main routine.

If the number of sounds is less than the total number, the type of sound is checked on the basis of the data stored in the register [i.e., sound output data (data pertaining to any of the types of sounds shown in Figs. 58 through 60)] and by reference to the sound output data table (G04_3).

A determination is made as to whether the type of sound is an alarm sound request (G04_4), a restoration effective sound request (G04_5), a prioritized single sound output request (G04_6), or a termination sound (G04_7).

If the type of sound is an alarm sound request (YES is selected in G04_4), the prioritized-single-sound-output-status storage area "HIT_NUM" is cleared (G04_8), and all-channel-playback-stop processing (i.e., processing G06 to be

described in detail later) is performed (G04_9).

All-channel-forced-reset data are set in the register (G04_10), and SD_OUT sound output data transmission processing is performed (G04_11). The register (sound output data) is forcefully transformed into a real-time playback mode (G04_12). SD_OUT data transmission sound output processing is performed (G04_13), and processing returns to the main routine.

If the type of sound is a restoration effective sound request (YES is selected in G04_5), the restoration-effective-sound-output-status storage area "PLAY_NUM" is checked (G04_14). The data stored in the register (i.e., the sound output request data) are compared with the data stored in the restoration-effective-sound-output-status storage area "PLAY_NUM" (G04_15).

A determination is made as to whether or not the data are identical with each other (G04_16). If they are identical (i.e., the requested sound has already been output in the previous processing), processing returns to the main routine. In contrast, if the data differ from each other, the data are considered to be new sound output data. Hence, SD_OUT sound output data transmission processing (i.e., processing G05 to be described in detail later) is performed (G04_17). The restoration-effective-sound-output-status storage area "PLAY_NUM" is cleared (G04_18), and processing returns to the main routine.

If the type of sound is a prioritized single sound output request (YES is selected in G04_6), the prioritized-single-sound-output-status storage area "HIT_NUM" is checked (G04_19), and the data stored in the register (i.e., the sound data) are compared with the data stored in the prioritized-single-sound-output-status storage area "HIT_NUM"(G04_20).

A determination is made as to whether or not the data are identical with each other (G04_21). If the data differ from each other, the register (sound data) is forcefully transformed into a real-time playback mode (for this reason, the sound is called "prioritized" single sound) (G04_22). Further, SD_OUT sound output data

transmission processing (i.e., processing G05 to be described in detail later) is performed (G04_23). The prioritized-single-sound-output-status storage area "HIT_NUM" is cleared (G04_24), and processing returns to the main routine.

If the type of sound is an end sound request (YES is selected in G04_7), the restoration-effective-sound-output-status storage area "PLAY_NUM" is cleared (G04_25), and all-channel-playback-stop processing (i.e., processing G06 to be described in detail later) is performed (G04_9).

All-channel-forced-reset data are set in the register (G04_10), and SD_OUT sound output data transmission processing is performed (G04_11). The register (sound output data) is forcefully transformed into a real-time playback mode (G04_12). SD_OUT data transmission sound output processing is performed (G04_13), and processing returns to the main routine.

If the type of sound does not match any of the foregoing types, the restoration-effective-sound-output-status storage area "PLAY_NUM" is cleared (G0-4_26); the prioritized-single-sound-output-status storage area "HIT_NUM" is cleared (G04_27); the register (sound output data) is forcefully transformed into a real-time playback mode (G04_28); and SD_OUT data transmission sound output processing is performed (G04_29). Then, processing returns to the main routine.

<SD_OUT Sound Output Data Transmission Processing: G05>

Fig. 255 is a flowchart showing procedures for SD_OUT data transmission sound output processing.

As shown in Fig. 255, the processing is for causing actual output of sound from the speaker 25, by means of outputting data pertaining to sound effects to the sound-source IC 206.

In the processing, the memory contents in the register are compared with end data "OFFH" (G05_1), thereby determining whether transmission data are end data (G05_2). If the transmission data are end data, processing returns to the main routine.

If transmission data are not end data, the data are output until the transmission data become end data (G05_3). The next data are set in the register (G05_4).

Fig. 52 shows details of a command to be transmitted to the sound IC 206. The transmission command consists of four bytes in total. The type of sound and a channel to be used are set in the first byte. A sound playback level (sound level) is set in the second byte. Pan-pot settings (i.e., choice as to whether stereophonic sound is to be output from the left, the right, or both the left and right in a balanced manner) are set in the third byte. A specific phrase number is set in the fourth byte (see the sound output data table shown in Figs. 61 through 74 for individual sound output data sets).

<All-Channel-Playback-Stop processing: G06>

Fig. 256 is a flowchart showing procedures for all-channel-playback-stop processing.

As shown in Fig. 256, the processing is for stopping generation of sound effects.

In this processing, a CH1 playback stop command "CMD_QUIT+CH1" is set in the register (G06_1), and SD_OUT sound output data transmission processing (i.e., processing G05 mentioned above) is performed (G06_2). A channel to be subjected to processing is set to the next channel (G06_3).

A determination is made as to whether or not all channels have been subjected to the processing (G06_4). Processing pertaining to steps (G06_2 and G06_3) is repeated until all the channels finish undergoing processing.

By means of the foregoing processing, the sub-control board 200 controls the image control board 300 under control of the main control board 100, whereby an image is displayed on the image display section 13. Further, the sub-control board 200 controls the sound-source IC 206 under control of the main control board 100, whereby sound effects are produced from the speaker 25.

In the previous embodiment, the control unit is constituted of the main control board 100 situated in the highest hierarchical level, the sub-control board 200 situated in an intermediate hierarchical level, and the image control board 300 situated in a lower hierarchical level. The control board situated in a lower hierarchical level may be constituted of not only the image control board 300 but of a plurality of control boards specialized for other control processing operations.

Although the present invention has been described by taking a slot machine as a typical gaming machine, the present invention can be applied to other gaming machines, such as Pachinko machines, Pachislo gaming machines (a hybrid machine of a Pachinko machine and a slot machine), Arranged ball gaming machines, or JongKyu Gaming Machines (a hybrid of a MahJong game machine and a Pachinko machine).

The present invention is embodied as having the configurations mentioned previously and yields the following advantages.

A gaming machine according to the present invention comprises a main control device for controlling the first hierarchical level, which is the highest hierarchical processing level of the gaming machine; an intermediate control device for controlling the second hierarchical level situated lower than the first hierarchical level, under control of the main control device; and a lower control device for controlling a third hierarchical level situated lower than the second hierarchical level, under control of the intermediate control device.

Such a construction enables separation of control processing, thereby mitigating loads imposed on individual control means and enabling smooth control of the gaming machine. In the case of a plurality of gaming machines requiring different gaming procedures, control shared among the gaming machines can be delegated to the main control device. Control operations which change from machine to machine are delegated to the intermediate control device or the lower control device. As a result, the main control device can be shared among the

gaming machines, thereby diminishing manufacturing and maintenance costs.

Preferably, the control device performs processing for making a decision pertaining to the player's profits, and the lower control device controls image display to be performed by the image display device. Further, the intermediate control device preferably performs control operations other than those to be performed by the lower control device.

Image control processing which requires great processing capability and storage of an enormous amount of image data is delegated to the lower control device, thereby mitigating loads imposed on the main control device and those imposed on the intermediate control device. Therefore, control processing of the gaming machine can be performed more smoothly.

Preferably, the intermediate control device performs control operations including a sound generation control operation to be performed by the sound generation device.

The only requirement of the lower control device is to perform image control operations. Accordingly, control processing is decentralized, thereby enabling much smoother control processing of the gaming machine.

Preferably, the lower control device comprises a plurality of control boards specialized for specific control processing operations, respectively.

Consequently, the control processing required in the gaming machine can be decentralized further. In a case where specific control processing is shared among a plurality of gaming machines requiring different gaming procedures, a board for performing the specific control processing can be shared, thereby diminishing manufacturing and maintenance costs.